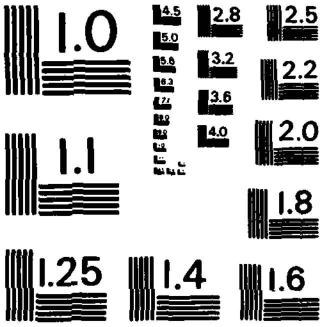
AD-A121 543 MICROPROCESSO IMPLEMENTATION OF OPTIONAL FUNCTIONS OF 1/1
SYNCHRONOUS BIT-OR..(U) DELTA INFORMATION SYSTEMS INC
JENKINTOWN PA S URBAN 19 MAY 82 NCS-TIB-82-3
UNCLASSIFIED DCA100-81-C-0025 F/G 17/2 NL

END

FILMED

DTIC


1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

3

NCS TIB 82-3

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 82-3

MICROPROCESSOR IMPLEMENTATION OF OPTIONAL FUNCTIONS OF SYNCHRONOUS BIT–ORIENTED DATA LINK CONTROL PROCEDURES

MAY 1982

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NCS-TIB 82-3	2. GOVT ACCESSION NO. AD-A121543	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Microprocessor Implementation of Optional Functions of Synchronous Bit-Oriented Data Link Control Procedures		5. TYPE OF REPORT & PERIOD COVERED FINAL
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Steve Urban		8. CONTRACT OR GRANT NUMBER(s) DCA100-81-C-0025
9. PERFORMING ORGANIZATION NAME AND ADDRESS Delta Information Systems, Inc. 259 Wyncote Road Jenkintown, PA 19046		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS National Communications System Office of Technology and Standards Washington, DC 20305		12. REPORT DATE May 19,1982
		13. NUMBER OF PAGES 73
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMEN" *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Telecommunications	Data Transmission Systems
Data Link	Protocols, Communications
Microprocessors	
ADCCP	

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report contains a block diagram, flow charts, and computer programming for the following optional functions in Federal Standard 1003 (Synchronous Bit Oriented Data Link Control Procedures): Address Extention Function for all three classes of procedures (Unbalanced Normal, Balanced Asynchronous, and Unbalanced Asynchronous); Reset Function for the Balanced Aysnchronous class of procedure only; Delete Command I Frame Function for all three classes of procedures; Delete Response I Frame Function for all three classes of procedures; unnumbered Polling Function for all three

classes of procedures.

NCS TECHNICAL INFORMATION BULLETIN 82-3

MICROPROCESSOR IMPLEMENTATION OF OPTIONAL FUNCTIONS
OF
SYNCHRONOUS BIT-ORIENTED DATA LINK CONTROL PROCEDURES

May 1982

PREPARED BY:

DELTA INFORMATION SYSTEMS, INC.
310 Cottman Street
Jenkintown, Pennsylvania 19046

APPROVED FOR PUBLICATION:

Marshall L Cain

MARSHALL L. CAIN
Assistant Manager
Office of NCS Technology
and Standards

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunication Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee identifies, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems or to the achievement of a compatible and efficient interface between computer and telecommunication systems. In developing and coordinating these standards a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the Electronic Industries Association, the American National Standards institute, the International Organization for Standardization, and the International Telegraph and Telephone Consultative Committee of the International Telecommunication Union. This Technical Information Bulletin presents an overview of an effort which is contributing to the development of compatible Federal, national, and international standards in the area of data link control procedures. It has been prepared to inform interested Federal activities of the progress of these efforts. Any comments, inputs or statements of requirements which could assist in the advancement of this work, are welcome and should be addressed to:

Office of the Manager
National Communications System
ATTN: NCS-TS (Frank McClelland)
Washington, D.C. 20305
(202) 692-2124

MICROPROCESSOR IMPLEMENTATION OF OPTIONAL FUNCTION OF SYNCHRONOUS BIT-ORIENTED DATA LINK CONTROL PROCEDURES

Final Report

Submitted To:

NATIONAL COMMUNICATIONS SYSTEM
Office of Technology and Standards
Washington, D.C. 20305

Contracting Agency:

DEFENSE COMMUNICATIONS AGENCY

Contract Number
DCA100-81-C-0025

May 19, 1982

Submitted By:

DELTA INFORMATIONSYSTEMS, INC.
310 Cottman Street
Jenkintown, Pennsylvania 19046

TABLE OF CONTENTS

ILLUSTRATIONS

1.0 INTRODUCTION

This document summarizes the work performed by Delta Information Systems, Inc. for the Office of Technology and Standards of the National Communications System, an organization of the U.S. Government, under Purchase Order DCA100- 81-C-0025. The Office of Technology and Standards, headed by National Communications System Assistant Manager Marshall L. Cain, is responsible for the management of the Federal Telecommunications Standards Program, which develops telecommunication standards whose use is mandatory by all Federal agencies. The objective of this program is to develop a block diagram, flow charts, and computer programming for the following tasks in accordance with Federal Standard 1003.

Address Extention Function for all three classes of procedures (Unbalanced Normal, Balanced Asynchronous, and Unbalanced Asynchronous).

- Reset Function for the Balanced Aysnchronous class of procedure only.
- Delete Command I Frame Function for all three classes of procedures.
- Delete Response I Frame Function for all three classes of procedures.
- Unnumbered Polling Function for all three classes of procedures.

- Initialization Function for all three classes of procedures.
- Unnumbered Information Function for all three classes of procedures.

The purpose of this effort is to determine the feasibility of using the M6800 or similar microprocessor to implement this type of protocol, and to obtain an estimate of memory and processor resources that would be required. The Office of Technology and Standards will use the information to advise other Federal agencies who implement the standard and, when merged with the results of other studies, to evaluate the operational and economic impact of incorporating various options in Federal Standard 1003.

The effort necessarily has focussed on the software required to implement the protocol itself, and is by no means a total hardward/software system design that would be required to develop a complete system. Complete system development is, of course, beyond the scope of this program.

Section 2 of this report contains a discussion of the method of implementation for the seven listed options and a list of state variables and parameters. Sections 3 through 8 include flow charts, code and a discussion of memory requirements and throughput for each of the options. The code was assembled on a 6800 cross-assembler and tested on a 6800 microcomputer supplied by Delta Information Systems.

2.0 SYSTEM DESIGN CONSIDERATIONS

The block diagram in Figure 2-1 shows a link with one primary/combined and one secondary/combined station communicating with each other by sending information in both directions. That is, either station may be a source or sink of data or both. Two-way simultaneous transmission is assumed. Although many secondary stations may communicate with one primary station, the objectives of this program can be met with no loss of generality, by assuming the existence of only one secondary station.

Each station, primary, secondary, or combined is made up of a microcomputer, an LSI interface to the link, and a user which supplies and uses the data to be communicated. The primary and secondary stations are physcially very similar; operationally, of course, the primary must supervise and control a number of secondary stations, and thus it requires a larger data structure and somewhat more complicated code.

For the purpose of this program, the microcomputer can be assumed to be very basic-microprocessor, memory (RAM and ROM), interface chips, clock, etc. A discussion of the interface chips, operating system considerations and general design features may be found in a previous report.(1)

The objective of this effort is to determine the incremental change in the number of instructions and processor time required for each of seven optional functions listed above, implemented on a Motorola 6800 microprocessor. These

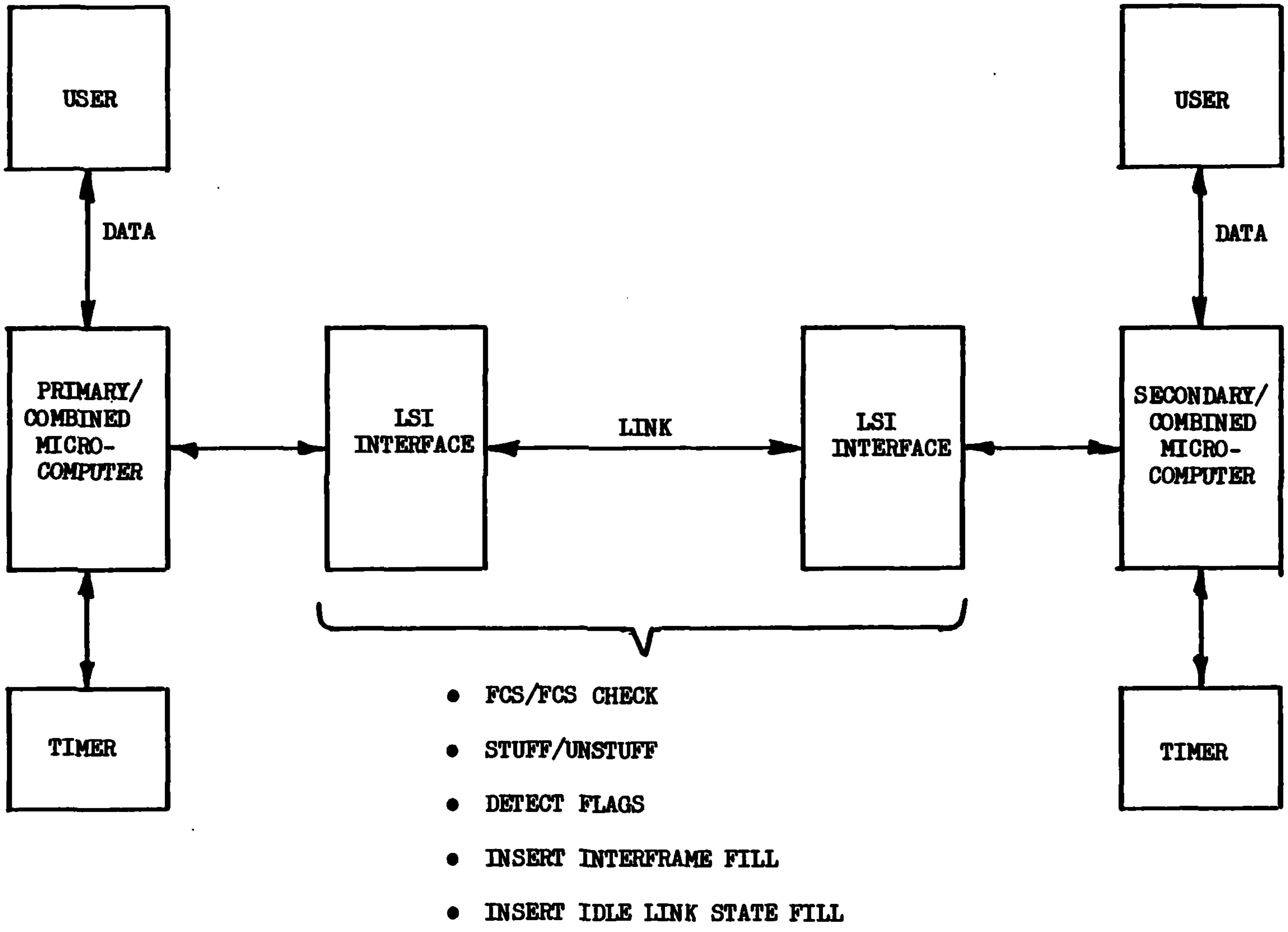


Figure 2-1 System Block Diagram

optional functions are achieved by the addition, or deletion, of commands and responses with respect to those present in one of the three basic classes of procedures.

No attempt has been made to produce a single basic design to accomodate all of the options one at a time or in combinations; in other words, each option is implemented starting from the same previously designed baseline so that the effect on memory requirements and throughput can be evaluated for each option.

Detailed flow charts and code for each option are compared with those of the baseline system to obtain the difference in memory requirements and throughput.

Those state variables and other parameters that are used by more than one routine and included in the code in the following sections are defined in Figure 2-2. A discussion of these may be found in Reference 1. Two of the flow charts in this previous report required some minor changes. These are included in Figures 2-3 and 2-4.

```
ERR LINE  ADDR  B1 B2 B3 B4        ADCCP MAIN                                                    PAGE    1

      1                                    TITLE   'ADCCP MAIN'
      2                                    LIST    X
      3                                    XREF    I,RR,RNR,REJ,SREJ,SNRM,UA,DISC,DM,FRMR,RSET
      4                                    XREF    SARM,SABM,UP,SIM,RIM,UI
      5                            *
      6                            *** DEFINE 6856 PROTOCOL CHIP ADDRESSES
      7                            *
      8  EC25                      MCR     EQU     $EC25     MODE CONTROL REGISTER
      9  EC24                      SAR     EQU     $EC24     SECONDARY ADDRESS REGISTER
     10  EC23                      TCR     EQU     $EC23     TRANSMITTER CONTROL
     11  EC22                      TDB     EQU     $EC22     TRANSMITTER DATA BUFFER
     12  EC27                      RCR     EQU     $EC27     RECEIVER CONTROL REGISTER
     13  EC26                      TSR     EQU     $EC26     TRANSMITTER STATUS REGISTER
     14  EC21                      RSR     EQU     $EC21     RECEIVER STATUS REGISTER
     15  EC20                      RDB     EQU     $EC20     RECEIVED DATA BUFFER
     16                            *

     18                            **
     19                            **   RAM VARIABLE STORAGE
     20                            **
     21                            *
     22                            *   BASE SECTION TABLES
     23                            *
     24                                 BSCT
     25  0000  02 04 06 09         ADDR    FCB    2,4,6,9   UNIQUE STATION ADDRESS
     26  0004                              RMB    8         SPACE FOR 2 ADDRESSES
     27                            *
     28                            ** JUMP TABLE FOR RECEIVED COMM/RESP
     29                            *
     30  000C  00 00          E    JMPTAB  FDB     I
     31  000E  00 00          E            FDB     RR
     32  0010  00 00          E            FDB     RNR
     33  0012  00 00          E            FDB     REJ
     34  0014  00 00          E            FDB     SREJ
     35  0016  00 00          E            FDB     SNRM
     36  0018  00 00          E            FDB     UA
     37  001A  00 00          E            FDB     DISC
     38  001C  00 00          E            FDB     DM
     39  001E  00 00          E            FDB     FRMR
     40  0020  00 00          E            FDB     RSET
     41  0022  00 00          E            FDB     SARM
     42  0024  00 00          E            FDB     SABM
     43  0026  00 00          E            FDB     UP
     44  0028  00 00          E            FDB     SIM
     45  002A  00 00          E            FDB     RIM
     46  002C  00 00          E            FDB     UI
```

Figure 2-2. Variables and Parameters

```
ERR LINE  ADDR  B1 B2 B3 B4         ADCCP MAIN                                                      PAGE    2

       48                          *
       49                          **   STATES/MODES
       50                          *
       51  002E                    STAT     RMB    1       STATION TYPE AND MODE:
       52                          *                          BIT 8 - UN/PRI
       53                          *                          BIT 7 - UN/SEC
       54                          *                          BIT 6 - UA/PRI
       55                          *                          BIT 5 - UA/SEC
       56                          *                          BIT 4 - BA
       57  002F                    OPSTAT   RMB    1       OPERATIONAL STATE:
       58                          *                          -2=FRMR
       59                          *                          -1=IS
       60                          *                           0=LDS
       61                          *                           1=ITS
       62  0030                    ITSMOD   RMB    1       ITS MODE:
       63                          *                           0=NRM
       64                          *                           1=ARM
       65                          *                           2=ADM
       66  0031                    ISMOD    RMB    1       IS MODE
       67                          *
       68  0032                    LDSMOD   RMB    1       LDS MODE:
       69                          *                           0=NDM
       70                          *                           1=ADM
       71  0033                    REMBUS   RMB    1       REMOTE BUSY
       72                          *                           1=TRUE
       73                          *                           0=FALSE
       74  0034                    STABUS   RMB    1       STATION BUSY
       75                          *                           1=TRUE
       76                          *                           0=FALSE
       77                          *
       78                          *** OTHER VARIABLES
       79                          *
       80  0035                    LENGMX   RMB    1       MAX LENGTH OF I-FRAME MESS (BYTES)
       81  0036                    PBIT     RMB    1       POLL BIT
       82  0037                    FBIT     RMB    1       FINAL BIT
       83  0038                    S        RMB    1       SEND VARIABLE
       84  0039                    R        RMB    1       RECEIVE VARIABLE
       85  003A                    NS       RMB    1       SEND SEQUENCE NUMBER
       86  003B                    NR       RMB    1       RECEIVE SEQUENCE NUMBER
       87  003C                    RDAFLG   RMB    1       RECEIVE DATA AVAILABLE FLAG
       88  003D                    RDBUFF   RMB    1       RECEIVE DATA BUFFER
       89  003E                    CNTFLD   RMB    1       CONTROL FIELD RECEIVED
       90  003F                    FTYPE    RMB    1       FRAME TYPE
       91  0040                    POLLP    RMB    1       PROVISIONAL POLL BIT
       92  0041                    NRP      RMB    1          "       RECEIVED SEQUENCE NUMBER
       93  0042                    NSP      RMB    1          "       SEND          "       "
       94  0043                    CFIELD   RMB    1       OUTPUT CONTROL BYTE
       95  0044                    UPFLAG   RMB    1       UP RECEIVED
       96                          *                           1=TRUE
       97                          *                           0=FALSE
       98  0045                    DAVAIL   RMB    1       DATA AVAILABLE FLAG
       99                          *                           1=TRUE
      100                          *                           0=FALSE
```

Figure 2-2. Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4        ADCCP MAIN

     101  0046                    LFRN     RMB     1        LAST FRAME
     102                          *                            1=TRUE
     103                          *                            0=FALSE
     104  0047                    GBFLG    RMB     1        RECEIVED BYTE VALIDITY
     105                          *                            -2=EOM/FCS ERROR
     106                          *                            -1=EOM/ABORT
     107                          *                             0=NORMAL
     108                          *                             1=EOM
     109  0048                    FRMBIF   RMB     3        FRMR BASIC INFORMATION FIELD
     110  004B                         END

  ASSEMBLER ERRORS =     0
```

Figure 2-2. Cont.

```
                              CROSS REFERENCE

LABEL     VALUE         REFERENCE

ADDR    B 0000             -25
CFIELD  B 0043             -94
CNTFLD  B 003E             -89
DAVAIL  B 0045             -98
DISC    E 0007               3     37
DM      E 0008               3     38
FBIT    B 0037             -82
FRMDIF  B 0048            -109
FRMR    E 0009               3     39
FTYPE   B 003F             -90
GBFLG   B 0047            -104
I       E 0000               3     30
ISMOD   B 0031             -66
ITSMOD  B 0030             -62
JMPTAB  B 000C             -30
LDSMOD  B 0032             -68
LENGMX  B 0035             -80
LFRM    B 0046            -101
MCR       EC25              -8
MEMORY  M 0000               0
NARG      0000               0
NR      B 003B             -86
NRP     B 0041             -92
NS      B 003A             -85
NSP     B 0042             -93
OPSTAT  B 002F             -57
PBIT    B 0036             -81
POLLP   B 0040             -91
R       B 0039             -84
RCR       EC27             -12
RDAFLG  B 003C             -87
RDD       EC20             -15
RDBUFF  B 003D             -88
REJ     E 0003               3     33
RENDUS  B 0033             -71
RIM     E 000F               4     45
RNR     E 0002               3     32
RR      E 0001               3     31
RSET    E 000A               3     40
RSR       EC21             -14
S       B 0038             -83
SABM    E 000C               4     42
SAR       EC24              -9
SARM    E 000B               4     41
SIM     E 000E               4     44
SNRM    E 0005               3     35
SREJ    E 0004               3     34
STABUS  B 0034             -74
STACK   S 0000               0
```

Figure 2-2. Variables and Parameters

```
STAT     D 002E            -51
TCR        EC23            -10
TDB        EC22            -11
TSR        EC26            -13
UA       E 0006              3    36
UI       E 0010              4    46
UP       E 000D              4    43
UPFLAG   D 0044            -95
```

Figure 2-2. Cont.

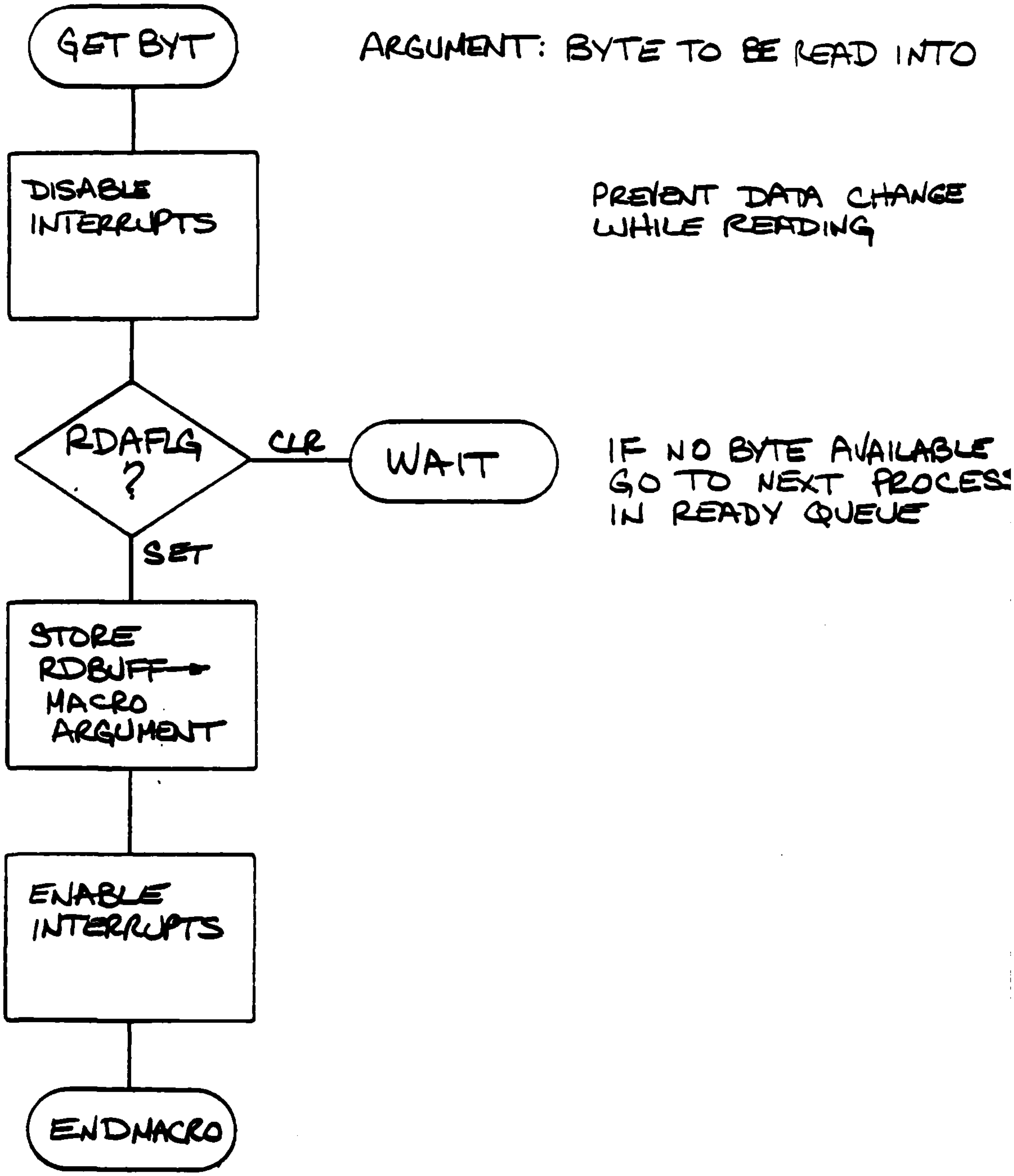

GET BYT
ARGUMENT: BYTE TO BE READ INTO
DISABLE INTERRUPTS
PREVENT DATA CHANGE WHILE READING
RDAFLG ?
CLR
WAIT
IF NO BYTE AVAILABLE GO TO NEXT PROCESS
IN READY QUEUE
SET
STORE RDBUFF → MACRO ARGUMENT
ENABLE INTERRUPTS
ENDMACRO

Figure 2-3. Read Data Byte Macro.

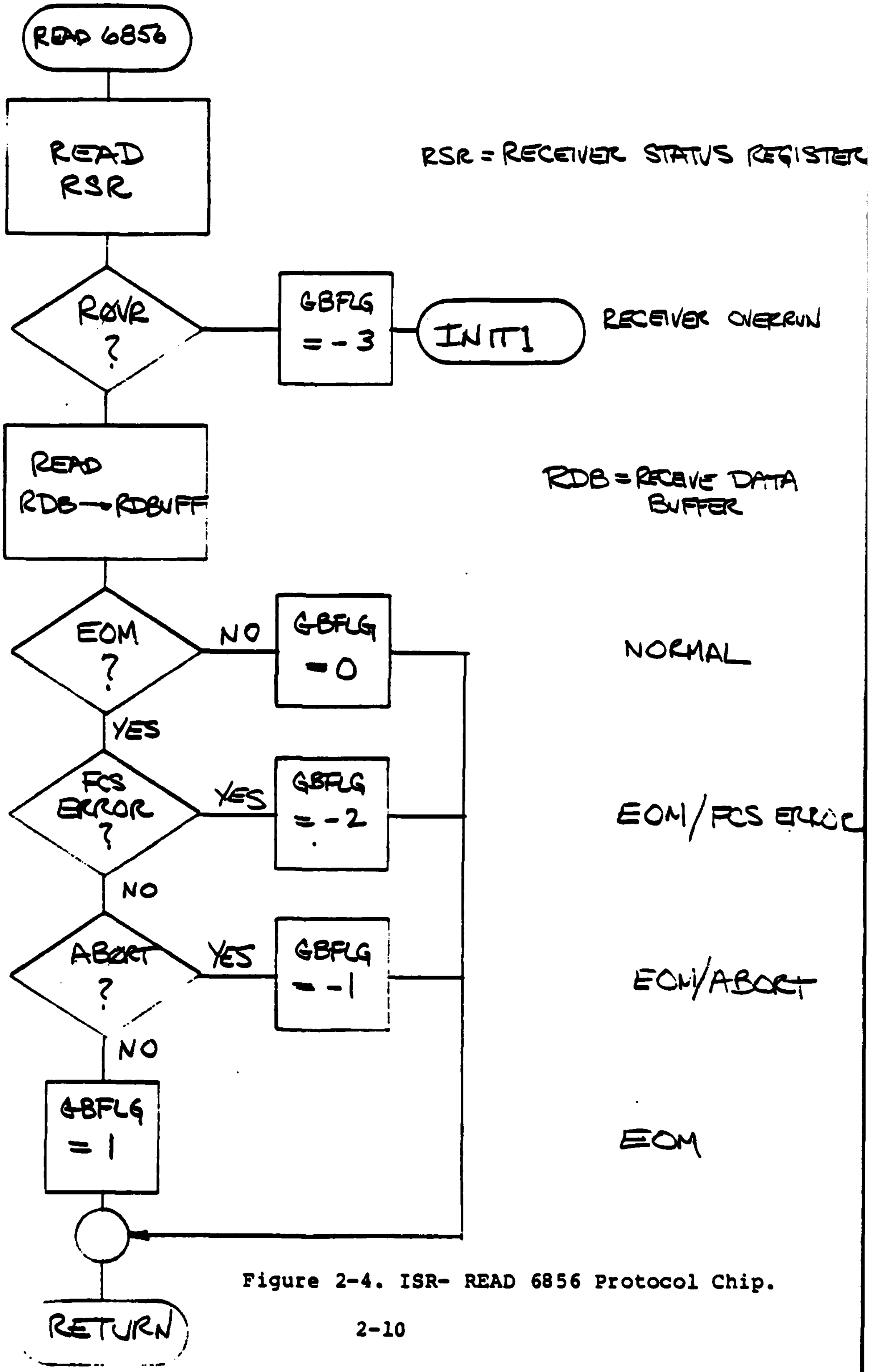


Figure 2-4. ISR- READ 6856 Protocol Chip.

3.0 ADDRESS EXTENSION FUNCTION

This option provides for greater than single octet addressing by means of the Extended Address Format. The extended format provides an address field which is made up of a sequence of octets, each having a "0" (Zero) as the first bit of the octet except for the last octet which has a "1" in the first bit position.

Processing of the received address is accomplished in the Receive Process (RCV). The flow chart for the RCV process is shown in Figure 3-1. A major subroutine called by RCV, the RCNTRL subroutine which processes the control field, is given in Figure 3-2. An expanded flow chart of the extended address handler is given in Figure 3-3. The 6800 assembly language code for the RCV process and the RCNTRL subroutine is presented in Figures 3-4 and 3-5 respectively.

The number of instructions required to perform the extended address can be estimated by examining Figure 3-4. The code for address processing is included in lines 52 through 132. Examination of this code shows that few (less than ten) additional instructions are required to perform the extended addressing function as opposed to single octet addressing. The extra processing time required to handle the extended address is negligible; however, there is a minor effect on throughput due to the increase in message length. The effect is very small for I/UI frames and somewhat larger for supervisory and other unnumbered frames.

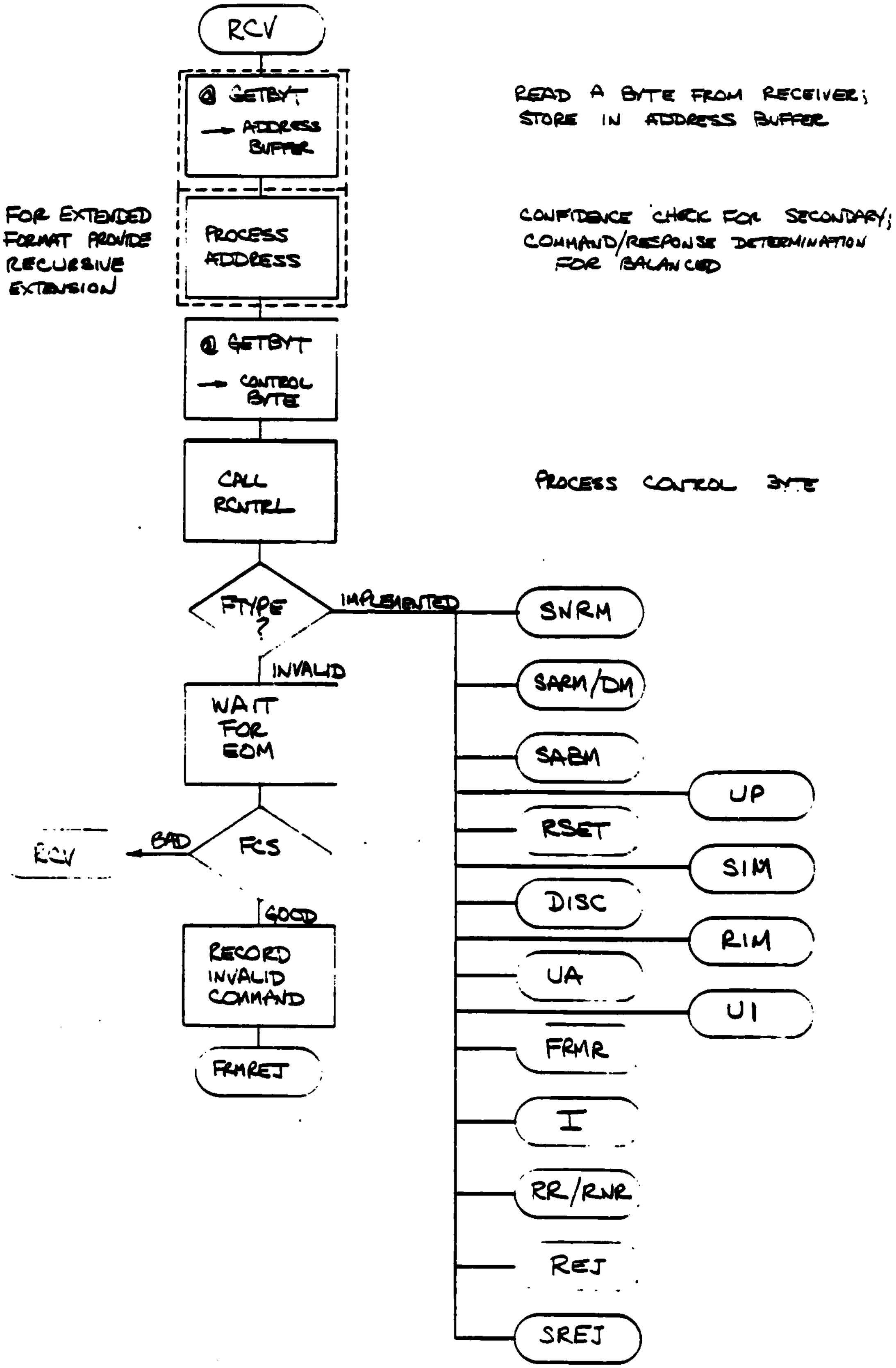


Figure 3-1 RCV Process

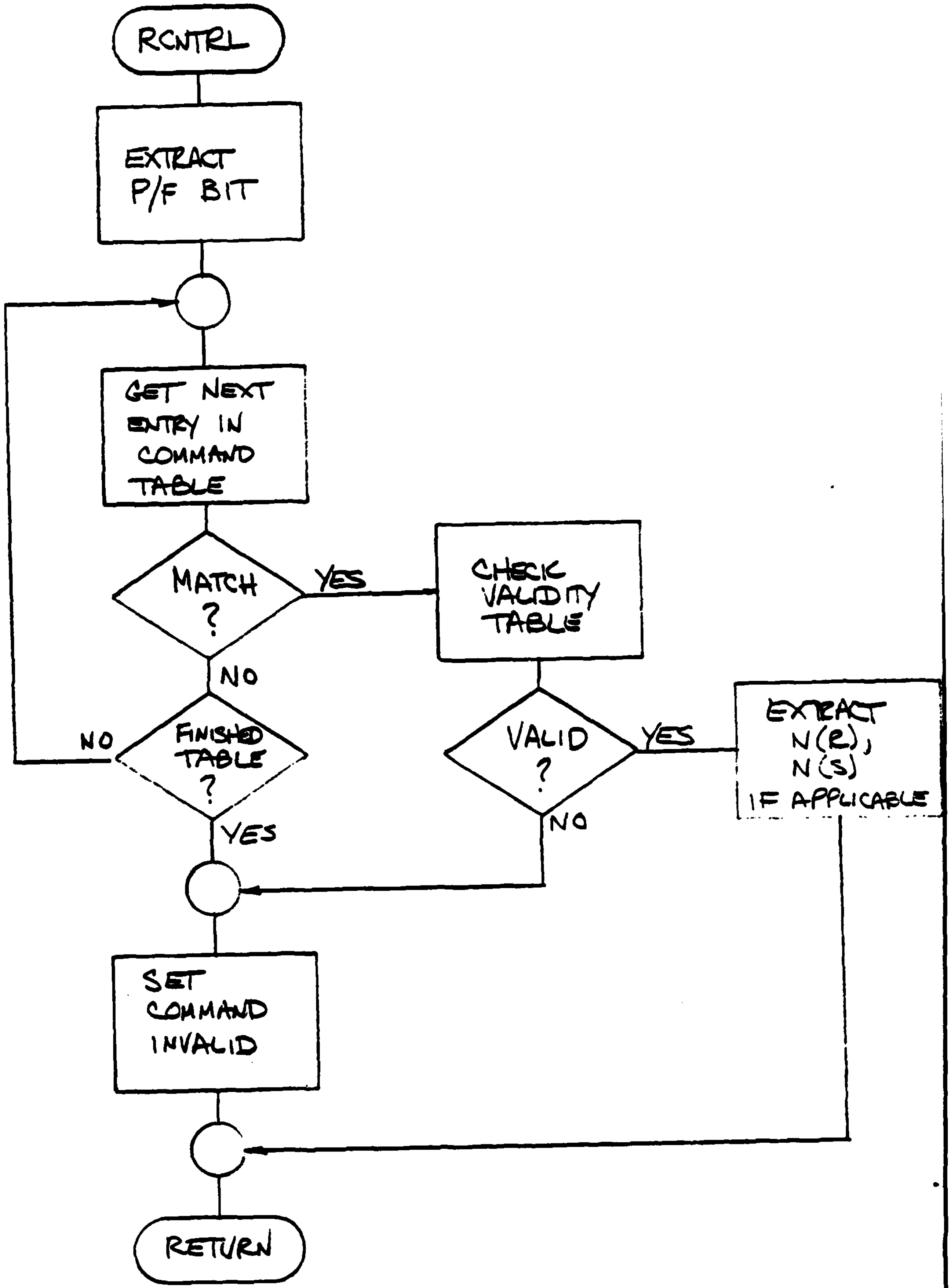


Figure 3-2 RCNTRL Subroutine

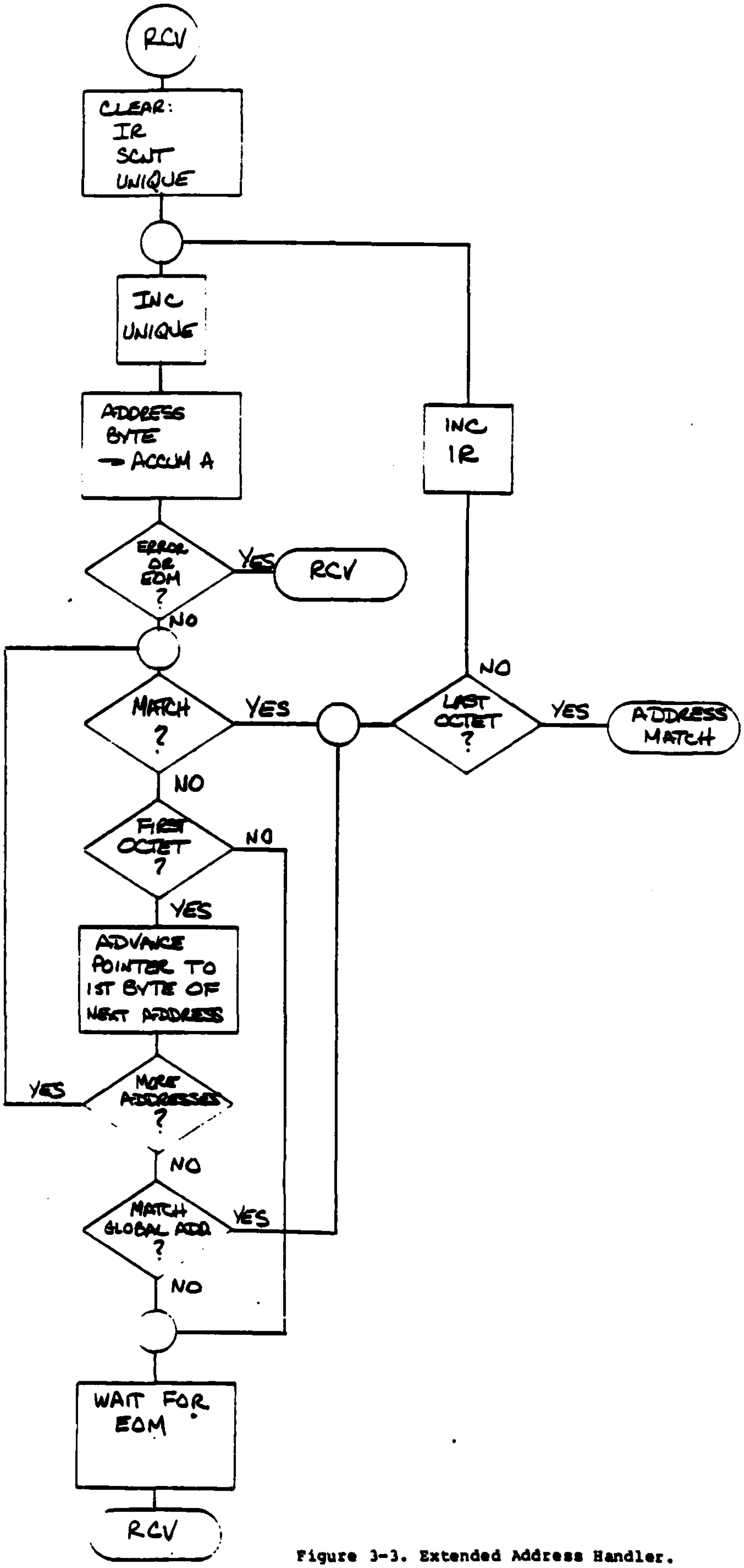
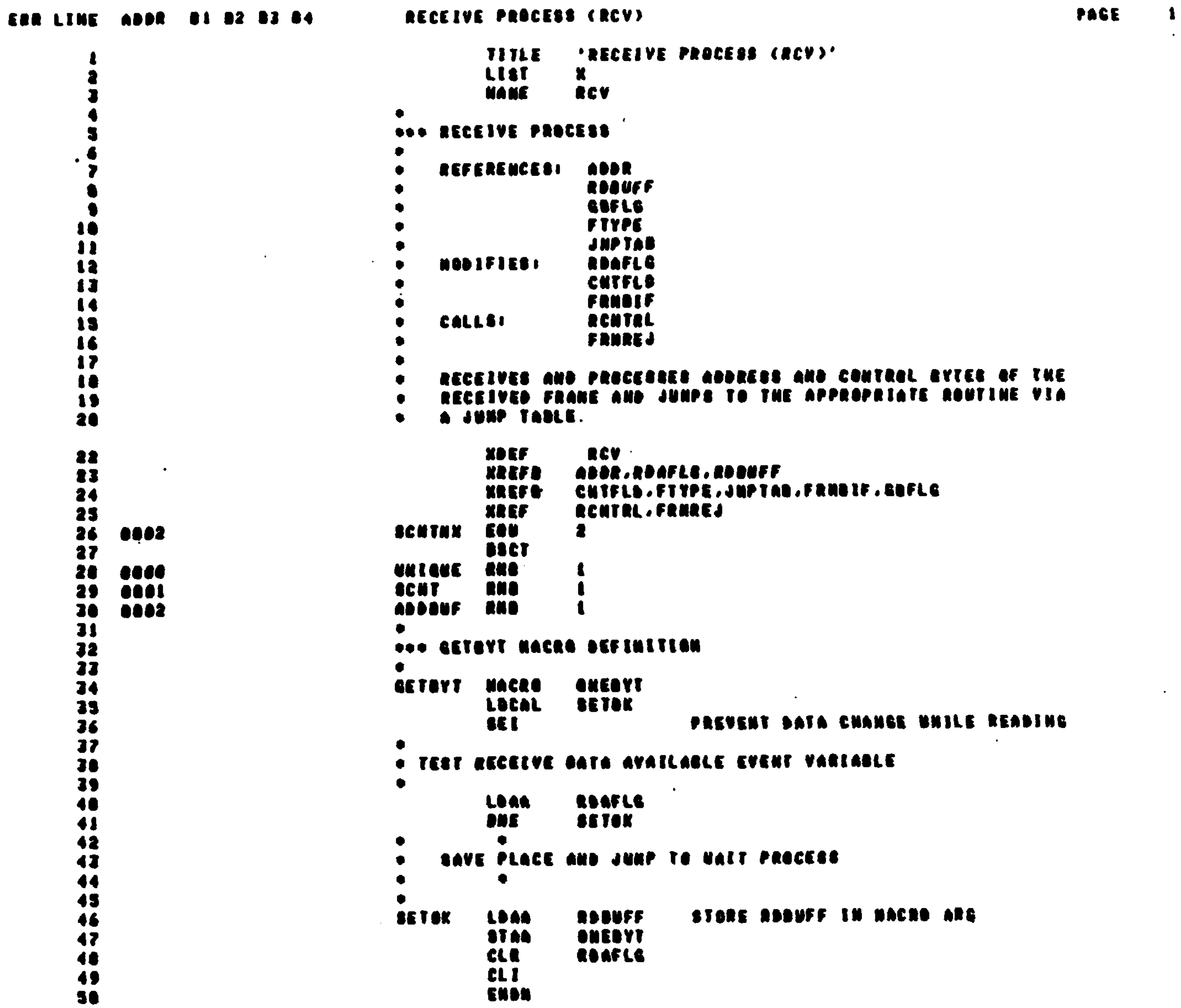


Figure 3-3. Extended Address Handler.



```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE PROCESS (RCV)                                      PAGE    1

       1                                   TITLE   'RECEIVE PROCESS (RCV)'
       2                                   LIST    X
       3                                   NAME    RCV
       4                           *
       5                           *** RECEIVE PROCESS
       6                           *
       7                           *   REFERENCES:  ADDR
       8                           *                RDBUFF
       9                           *                GUFLG
      10                           *                FTYPE
      11                           *                JMPTAB
      12                           *   MODIFIES:    RDAFLG
      13                           *                CNTFLG
      14                           *                FRMBIF
      15                           *   CALLS:       RCNTRL
      16                           *                FRMREJ
      17                           *
      18                           *   RECEIVES AND PROCESSES ADDRESS AND CONTROL BYTES OF THE
      19                           *   RECEIVED FRAME AND JUMPS TO THE APPROPRIATE ROUTINE VIA
      20                           *   A JUMP TABLE.

      22                                   XDEF    RCV
      23                                   XREFB   ADDR,RDAFLG,RDBUFF
      24                                   XREFB   CNTFLG,FTYPE,JMPTAB,FRMBIF,GUFLG
      25                                   XREF    RCNTRL,FRMREJ
      26  0002                     SCNTMX  EQU     2
      27                                   BSCT
      28  0000                     UNIQUE  RMB     1
      29  0001                     SCNT    RMB     1
      30  0002                     ADDBUF  RMB     1
      31                           *
      32                           *** GETBYT MACRO DEFINITION
      33                           *
      34                           GETBYT  MACRO   ONEBYT
      35                                   LOCAL   SETOK
      36                                   SEI                PREVENT DATA CHANGE WHILE READING
      37                           *
      38                           * TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
      39                           *
      40                                   LDAA    RDAFLG
      41                                   BNE     SETOK
      42                           *        *
      43                           *   SAVE PLACE AND JUMP TO WAIT PROCESS
      44                           *        *
      45                           *
      46                           SETOK   LDAA    RDBUFF     STORE RDBUFF IN MACRO ARG
      47                                   STAA    ONEBYT
      48                                   CLR     RDAFLG
      49                                   CLI
      50                                   ENDM
```

Figure 3-4. Receive Process Code

```
ERR LINE  ADDR  B1 B2 B3 B4       RECEIVE PROCESS (RCV)                                           PAGE    2

     52                                       PSCT
     53                              *
     54                              *    PROCESS ADDRESS
     55                              *
     56   0000  CE 00 00             RCV      LDX     #0
     57   0003  4F                            CLRA
     58   0004  97 01          B              STAA    SCNT
     59   0006  97 00          B              STAA    UNIQUE
     60   0008  7C 00 00       B     STADD    INC     UNIQUE
     61   000B                                GETBYT  ADDBUF
     62   000B  0F                  +         SEI                 PREVENT DATA CHANGE WHILE READING
     63                             +*
     64                             +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
     65                             +*
     66   000C  96 00          E    +         LDAA    RDAFLG
     67   000E  26 00               +         BNE     .00000
     68                             +*        *
     69                             +*   SAVE PLACE AND JUMP TO WAIT PROCESS
     70                             +*        *
     71                             +*
     72   0010  96 00          E    +.00000    LDAA    RDBUFF     STORE RDBUFF IN MACRO ARG
     73   0012  97 02          B    +         STAA    ADDBUF
     74   0014  7F 00 00       E    +         CLR     RDAFLG
     75   0017  0E                  +         CLI
     76   0018  7D 00 00       E              TST     GBFLG
     77   001B  2D E3                         BLT     RCV         ERROR
     78   001D  2E E1                         BGT     RCV         EOM
     79   001F  E6 00          E     STADD1   LDAB    ADDR,X      MATCH STATION ADDRESS?
     80   0021  11                            CBA
     81   0022  27 2A                         BEQ     AMATCH
     82   0024  D6 00          B              LDAB    UNIQUE      NO MATCH; CHECK FOR FIRST OCTET
     83   0026  5A                            DECB
     84   0027  26 11                         BNE     WEOM        NOT FIRST OCTET
     85   0029  08                            INX
     86   002A  08                            INX
     87   002B  08                            INX
     88   002C  08                            INX
     89   002D  D6 01          B              LDAB    SCNT
     90   002F  5C                            INCB
     91   0030  D7 01          B              STAB    SCNT
     92   0032  D1 03                         CMPB    SCNTMX+1
     93   0034  2F E9                         BLE     STADD1
     94   0036  81 FF                         CMPA    #$FF        NOT A VALID ADDRESS
     95   0038  27 1F                         BEQ     LOCTET      MATCHES GLOBAL ADDRESS
     96                              *
     97                              *    NULL ADDRESS OR NO MATCH
     98                              *

    100   003A                       WEOM     GETBYT  ADDBUF      NO MATCH  WAIT FOR EOM
    101   003A  0F                  +         SEI                 PREVENT DATA CHANGE WHILE READING
    102                             +*
    103                             +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
```

Figure 3-4. Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE PROCESS (RCV)

    104                        **
    105  003B  96 00        E  *          LDAA    RDAFLG
    106  003D  26 00           *          BNE     .00001
    107                        **          *
    108                        **    SAVE PLACE AND JUMP TO WAIT PROCESS
    109                        **          *
    110                        **
    111  003F  96 00        E  *.00001   LDAA    RDBUFF     STORE RDBUFF IN MACRO ARG
    112  0041  97 02        E  *          STAA    ADDBUF
    113  0043  7F 00 00     E  *          CLR     RDAFLG
    114  0046  0E              *          CLI
    115  0047  7D 00 00     E             TST     GDFLG
    116  004A  27 EE                      BEQ     WEOM
    117  004C  20 B2                      BRA     RCV

    119  004E  46                AMATCH  RORA                LOOK AT LEAST SIG BIT
    120  004F  25 08                      BCS     LOCTET     HAVE LAST OCTET OF ADDRESS
    121  0051  08                         INX
    122  0052  8C 00 0D                   CPX     #SCNTHX+4+5    SAFETY CHECK
    123  0055  2C A9                      BGE     RCV
    124  0057  20 AF                      BRA     STADD

    126  0059                    LOCTET
    127                            *      *
    128                            *   CHECK ADDRESS FOR CONSISTENCY WITH CONFIGURATION
    129                            *
    130                            *   SECONDARY - SCNT=0 OR SCNTHX+1
    131                            *   PRIMARY   - 0<SCNT<SCNTHX+1
    132                            *      *
    133                            *
    134                            *   PROCESS CONTROL BYTE
    135                            *
    136  0059                             GETBYT  CNTFLD
    137  0059  0F                 *       SEI                 PREVENT DATA CHANGE WHILE READING
    138                          **
    139                          ** TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
    140                          **
    141  005A  96 00        E  *          LDAA    RDAFLG
    142  005C  26 00           *          BNE     .00002
    143                          **        *
    144                          **  SAVE PLACE AND JUMP TO WAIT PROCESS
    145                          **        *
    146                          **
    147  005E  96 00        E  *.00002   LDAA    RDBUFF     STORE RDBUFF IN MACRO ARG
    148  0060  97 00        E  *          STAA    CNTFLD
    149  0062  7F 00 00     E  *          CLR     RDAFLG
    150  0065  0E              *          CLI
    151  0066  7D 00 00     E             TST     GDFLG
    152  0069  2D 95                      BLT     RCV        ERROR
    153  006B  2E 93                      BGT     RCV        EOM
    154  006D  BD 00 00     E             JSR     RCNTRL
    155  0070  7D 00 00     E             TST     FTYPE      FRAME VALID?
```

Figure 3-4. Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE PROCESS (RCV)                                              PAGE    4

     156  0073  20 06                          BLT     WEOM1      NO
     157  0075  DE 00           E              LDX     FTYPE      YES  PROCESS APPROPRIATE COMM/RESP
     158  0077  EE FE           E              LDX     JMPTAB-2,X
     159  0079  6E 00                          JMP     0,X
     160                            *
     161                            *   FRAME INVALID
     162                            *
     163  007B                      WEOM1    GETBYT  ADDBUF      WAIT FOR EOM
     164  007B  0F                 +         SEI                 PREVENT DATA CHANGE WHILE READING
     165                           +*
     166                           +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
     167                           +*
     168  007C  96 00           E  +         LDAA    RDAFLG
     169  007E  26 00              +         BNE     .00003
     170                           +*         *
     171                           +*   SAVE PLACE AND JUMP TO WAIT PROCESS
     172                           +*         *
     173                           +*
     174  0080  96 00           E  +.00003   LDAA    RDBUFF      STORE RDBUFF IN MACRO ARG
     175  0082  97 02           E  +         STAA    ADDBUF
     176  0084  7F 00 00        E  +         CLR     RDAFLG
     177  0087  0E                 +         CLI
     178  0088  7D 00 00        E            TST     GBFLG
     179  008B  27 EE                        BEQ     WEOM1
     180  008D  2E 03                        BGT     EOM
     181  008F  7E 00 00        P            JMP     RCV         ERROR
     182                            *
     183                            *   EOM
     184                            *
     185  0092  96 00           E   EOM      LDAA    CNTFLD      RECORD INVALID COMMAND
     186  0094  97 00           E            STAA    FRMDIF
     187  0096  7E 00 00        E            JMP     FRMREJ
     188  0099                               END

 ASSEMBLER ERRORS =    0
```

Figure 3-4. Cont.

```
                                 RECEIVE PROCESS (RCV)                                  PAGE    5

                              CROSS REFERENCE
LABEL      VALUE         REFERENCE

ADDBUF   B 0002           -30    73   112   178
ADDR     E 0000            23    79
AMATCH   P 004E            81  -119
CNTFLD   E 0003            24   140   185
EOM      P 0092           100  -108
FRMDIF   E 0006            24   106
FRMREJ   E 0009            25   107
FTYPE    E 0004            24   155   157
GOFLG    E 0007            24    76   115   151   170
JMPTAB   E 0005            24   150
LOCTET   P 0039            95   120  -126
MEMORY   M 0000             0
MARG       0000             0
RCNTRL   E 0000            25   154
RCV      P 0000           -56    77    78   117   123   152   153   181
RDAFLG   E 0001            23    66    74   108   113   141   149   168
                          176
RDBUFF   E 0002            23    72   111   147   174
SCNT     B 0001           -29    58    89    91
SCNTMX     0002           -26    92   122
STACK    S 0000             0
STADD    P 0000           -60   124
STADD1   P 001F           -79    93
UNIQUE   B 0000           -28    59    60    82
VEOM     P 003A            84  -100   116
VEOM1    P 007B           156  -163   179
```

Figure 3-4. Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4        RCNTRL SUBROUTINE                                        PAGE    1

      1                                       TITLE   'RCNTRL SUBROUTINE'
      2                                       LIST    X
      3                                       NAME    RCNTRL
      4                             *
      5                             *** RCNTRL SUBROUTINE
      6                             *
      7                             *   REFERENCES:  CNTFLD
      8                             *                CONTAB
      9                             *                VALTAB
     10                             *                FRTAB
     11                             *                STAT
     12                             *   MODIFIES:    POLLP
     13                             *                NRP
     14                             *                NSP
     15                             *                FTYPE
     16                             *
     17                             *   EXTRACTS PROVISIONAL P/F BIT, N(R), N(S) IF APPLICABLE.
     18                             *   DETERMINES FRAME TYPE.   CHECKS COMMAND/RESPONSE FOR
     19                             *   VALIDITY AGAINST MODE, STATION TYPE.

     21                                       XDEF    RCNTRL
     22                                       XREFS   CNTFLD,POLLP,NRP,NSP,FTYPE,STAT
     23                                       PSCT
     24  0000  96 00          E     RCNTRL    LDAA    CNTFLD
     25  0002  16                             TAB
     26  0003  C4 10                          ANDB    #$10
     27  0005  D7 00          E               STAB    POLLP        EXTRACT PROVISIONAL POLL BIT

     29  0007  CE 00 56       P               LDX     #CONTAB
     30  000A  16                             TAB
     31  000B  C4 01                          ANDB    #$01         TEST FOR I-FRAME FIRST
     32  000D  E1 00                          CMPB    0,X
     33  000F  27 20                          BEQ     MATCH
     34  0011  08                             INX                  NOT AN I-FRAME

     36  0012  16                             TAB
     37  0013  C4 0F                          ANDB    #$0F         TEST FOR SUPERVISORY FRAMES
     38  0015  E1 00                TEST1     CMPB    0,X
     39  0017  27 10                          BEQ     MATCH
     40  0019  08                             INX
     41  001A  8C 00 5B       P               CPX     #CONTAB+5
     42  001D  26 F6                          BNE     TEST1

     44  001F  16                             TAB                  NOT A SUPERVISORY FRAME
     45  0020  C4 EF                          ANDB    #$EF         TEST FOR UNNUMBERED FRAMES
     46  0022  E1 00                TEST2     CMPB    0,X
     47  0024  27 08                          BEQ     MATCH
     48  0026  08                             INX
     49  0027  8C 00 63       P               CPX     #CONTAB+13
     50  002A  26 F6                          BNE     TEST2
```

Figure 3-5. RCNTL Code

```
ERR LINE  ADDR  B1 B2 B3 B4        RCNTRL SUBROUTINE                                        PAGE    2

     51   002C  C6 FF             INVLD   LDAB    #-1         INVALID FRAME TYPE
     52   002E  D7 00         E           STAB    FTYPE
     53   0030  39                        RTS

     55   0031  D6 00         E   MATCH   LDAB    STAT        GET STATION TYPE AND MODE
     56   0033  E4 11                     ANDB    VALTAB-CONTAB,X      TEST AGAINST FRAME TYPE
     57   0035  27 F5                     BEQ     INVLD
     58   0037  E6 22                     LDAB    FRTAB-CONTAB,X
     59   0039  D7 00         E           STAB    FTYPE

     61   003B  C1 0A                     CMPB    #10
     62   003D  2E 16                     BGT     DONE         UNNUMBERED FRAME
     63   003F  16                        TAB
     64   0040  C4 E0                     ANDB    #$E0         GET N(R) PORTION
     65   0042  54                        LSRB
     66   0043  54                        LSRB
     67   0044  54                        LSRB
     68   0045  54                        LSRB
     69   0046  54                        LSRB
     70   0047  D7 00         E           STAB    NRP          STORE PROVISIONAL N(R)
     71   0049  D6 00         E           LDAB    FTYPE
     72   004B  C1 02                     CMPB    #2
     73   004D  26 06                     BNE     DONE         SUPERVISORY FRAME
     74   004F  16                        TAB
     75   0050  C4 0E                     ANDB    #$0E
     76   0052  54                        LSRB
     77   0053  D7 00         E           STAB    NSP          STORE PROVISIONAL N(S)
     78   0055  39                DONE    RTS

     80   0056  00                CONTAB  FCB     00000000B INFORMATION
     81   0057  01                        FCB     00000001B RECEIVE READY
     82   0058  05                        FCB     00000101B RECEIVE NOT READY
     83   0059  09                        FCB     00001001B REJECT
     84   005A  0D                        FCB     00001101B SELECTIVE REJECT
     85   005B  83                        FCB     10000011B SET NORMAL RESPONSE MODE
     86   005C  63                        FCB     01100011B UNNUMBERED ACKNOWLEDGEMENT
     87   005D  43                        FCB     01000011B DISCONNECT
     88   005E  0F                        FCB     00001111B DISCONNECTED MODE
     89   005F  87                        FCB     10000111B FRAME REJECT
     90   0060  8F                        FCB     10001111B RESET
     91   0061  0F                        FCB     00001111B SET ASYNC RESPONSE MODE
     92   0062  2F                        FCB     00101111B SET ASYNC BAL MODE
     93   0063  23                        FCB     00100011B UNNUMBERED POLLING
     94   0064  07                        FCB     00000111B SET INITIALIZATION MODE
     95   0065  07                        FCB     00000111B REQUEST INITIALIZATION MODE
     96   0066  03                        FCB     00000011B UNNUMBERED INFORMATION

     98   0067  F0                VALTAB  FCB     11110000B I
     99   0068  F0                        FCB     11110000B RR
    100   0069  F0                        FCB     11110000B RNR
```

Figure 3-5. Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4       RCNTRL SUBROUTINE                                      PAGE    3

     101  006A  F0                       FCB      11110000B REJ
     102  006B  F0                       FCB      11110000B SREJ
     103  006C  40                       FCB      01000000B SNRM
     104  006D  A8                       FCB      10101000B UA
     105  006E  58                       FCB      01011000B DISC
     106  006F  B8                       FCB      10111000B DM
     107  0070  A8                       FCB      10101000B FRMR
     108  0071  08                       FCB      00001000B RSET
     109  0072  B8                       FCB      10111000B SARM
     110  0073  08                       FCB      00001000B SABM
     111  0074  58                       FCB      01011000B UP
     112  0075  58                       FCB      01011000B SIM
     113  0076  A8                       FCB      10101000B RIM
     114  0077  F0                       FCB      11110000B UI

     116  0078  02 04 06 08       FRTAB  FCB      2,4,6,8,10,12,14,16,18,20,22,24,26
     117  007C  0A 0C 0E 10
     118  0080  12 14 16 18
     119  0084  1A
     120  0085  1C 1E 20 22              FCB      28,30,32,34
     121  0089                           END

  ASSEMBLER ERRORS =    0
```

Figure 3-5. Cont.

```
                          RCNTRL SUBROUTINE                                       PAGE    4

                       CROSS REFERENCE

LABEL      VALUE          REFERENCE

CNTFLD   E 0000             22     24
CONTAB   P 0056             29     41     49     56     50    -80
DONE     P 0055             62     73    -70
FRTAB    P 0070             50   -116
FTYPE    E 0004             22     52     59     71
INVLD    P 002C            -51     57
MATCH    P 0031             33     39     47    -55
MEMORY   M 0000              0
MARG       0000              0
NRP      E 0002             22     70
NSP      E 0003             22     77
POLLP    E 0001             22     27
RCNTRL   P 0000            -24
STACK    S 0000              0
STAT     E 0005             22     55
TEST1    P 0015            -30     42
TEST2    P 0022            -46     50
VALTAB   P 0067             56    -90
```

Figure 3-5. Cont.

4.0 DELETE RESET FUNCTION

This option removes the ability to reset the Send and Receive variables associated with only one direction of information flow by the deletion of the RSET command. This applies to the Balanced, Asynchronous class of procedures only.

Processing of the Reset function (RSET) is accomplished in the two routines RSET and TR-RSET used respectively for receive and transmit. Flow charts for these two routines are presented in Figures 4-1 and 4-2. Assembly language code for the receive RSET function is shown in Figure 4-3. If the RESET function is deleted, neither the receive nor transmit routines are required, resulting in a reduction of approximately 32 instructions (64 bytes) for receive and about 40 instructions for transmit. Since the transmission of the RSET command by a combined may be used to report an invalid N(R), some minor changes in the use of the FRMREJ subroutine may be implied. Effects on throughput are difficult to estimate at this level of implementation.

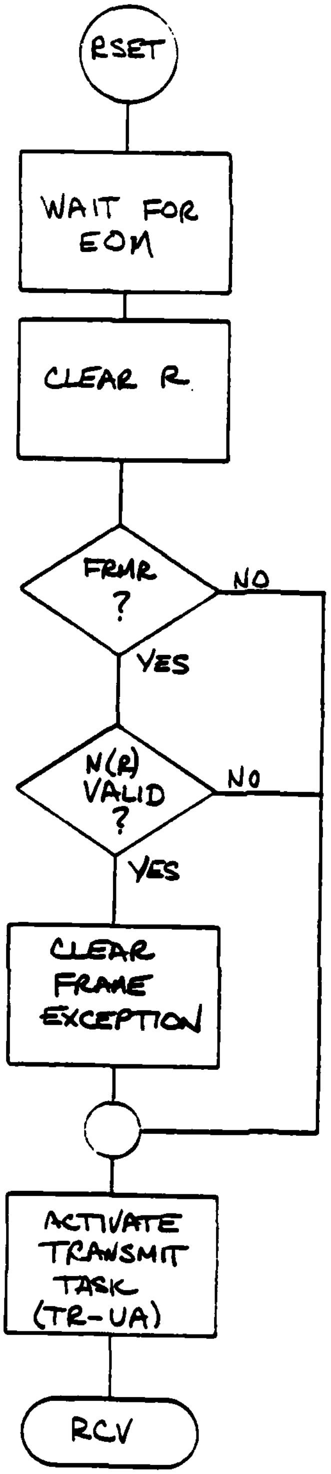


Figure 4-1. Receive RSET Command

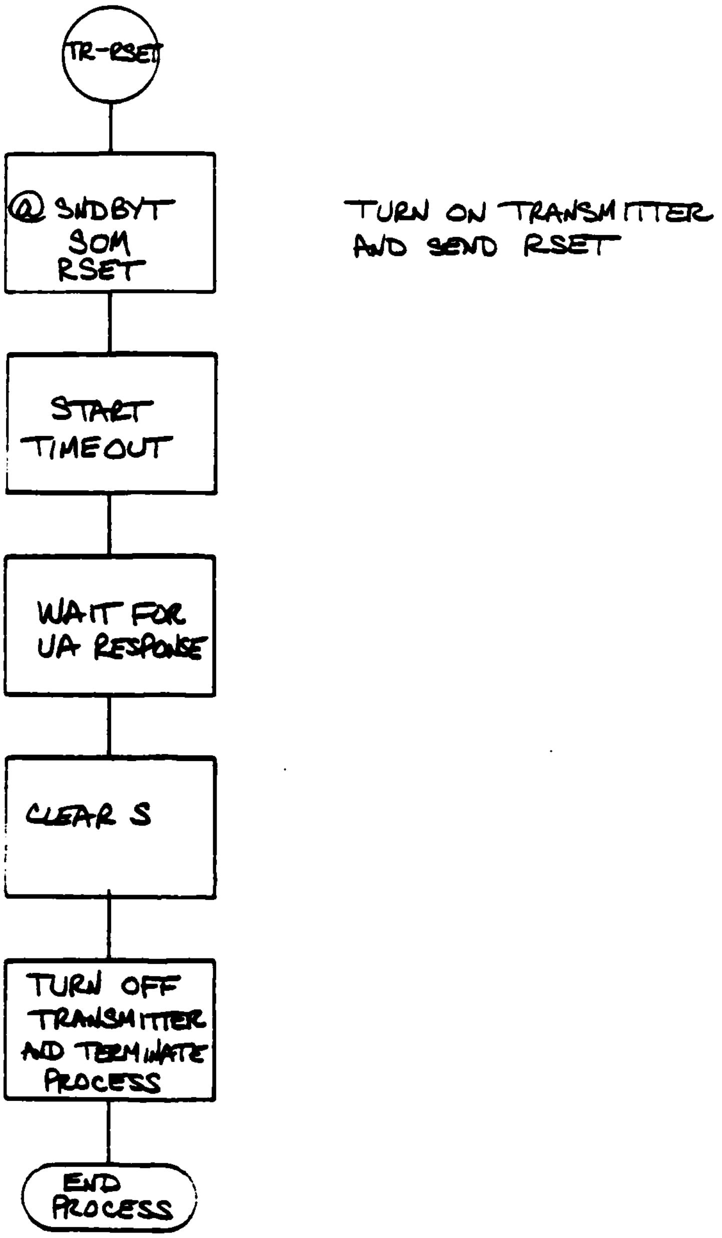


Figure 4-2. Transmit RSET

```
ERR LINE  ADDR  B1 B2 B3 B4       RECEIVE RSET COMMAND (RSET)                               PAGE    1
       1                                  TITLE   'RECEIVE RSET COMMAND (RSET)'
       2                                  LIST    X
       3                                  NAME    RSET
       4                          *
       5                          *** RECEIVE RSET COMMAND
       6                          *
       7                          *   REFERENCES:  GBFLG
       8                          *                FRMBIF
       9                          *   MODIFIES:    R
      10                          *                OPSTAT
      11                          *
      12                          *   CLEARS RECEIVE STATE VARIABLE (R) AND APPLICABLE FRMR
      13                          *   CONDITIONS IN ADDRESSED COMBINED STATION.  CONFIRMS
      14                          *   ACCEPTANCE OF RSET BY TRANSMITTING UA.

      16                                  XDEF     RSET
      17                                  XREFB   GBFLG,R,OPSTAT,FRMBIF
      18                                  XREFB   RDAFLG,RDBUFF
      19                                  XREF    RCV
      20                                  BSCT
      21  0000                    ADDBUF  RMB     1
      22                          *
      23                          *** GETBYT MACRO DEFINITION
      24                          *
      25                          GETBYT  MACRO   ONEBYT
      26                                  LOCAL   SETOK
      27                                  SEI               PREVENT DATA CHANGE WHILE READING
      28                          *
      29                          * TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
      30                          *
      31                                  LDAA    RDAFLG
      32                                  BNE     SETOK
      33                          *        *
      34                          *   SAVE PLACE AND JUMP TO WAIT PROCESS
      35                          *        *
      36                          *
      37                          SETOK   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
      38                                  STAA    ONEBYT
      39                                  CLR     RDAFLG
      40                                  CLI
      41                                  ENDM
      42                                  PSCT
      43                          *
      44                          *   WAIT FOR EOM
      45                          *
      46  0000                    RSET    GETBYT  ADDBUF
      47  0000  0F               +        SEI               PREVENT DATA CHANGE WHILE READING
      48                         +*
      49                         +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
      50                         +*
      51  0001  96 00         E  +        LDAA    RDAFLG
      52  0003  26 00            +        BNE     .00000
      53                         +*        *
```

Figure 4-3. Receive RSET Command Code

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE RSET COMMAND (RSET)                              PAGE    2

     54                            +*   SAVE PLACE AND JUMP TO WAIT PROCESS
     55                            +*         *
     56                            +*
     57   0005  96 00           E  +.00000   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
     58   0007  97 00           B  +         STAA    ADDBUF
     59   0009  7F 00 00        E  +         CLR     RDAFLG
     60   000C  0E                 +         CLI
     61   000D  7D 00 00        E            TST     GBFLG
     62   0010  2E 03                        BGT     EON      HAVE EON
     63   0012  7E 00 00        E            JMP     RCV      ERROR
     64   0015  7F 00 00        E   EON      CLR     R
     65   0018  96 00           E            LDAA    OPSTAT   FRMR?
     66   001A  8B 02                        ADDA    02
     67   001C  26 09                        BNE     NOTVAL   NOT FRMR
     68   001E  96 02           E            LDAA    FRMBIF+2
     69   0020  84 00                        ANDA    #$00
     70   0022  26 03                        BNE     NOTVAL   N(R) NOT VALID
     71   0024  7F 00 00        E            CLR     OPSTAT
     72   0027                     NOTVAL
     73                            *         *
     74                            *    ACTIVATE TRANSMIT TASK
     75                            *         *
     76   0027  7E 00 00        E            JMP     RCV
     77   002A                               END

 ASSEMBLER ERRORS =    0
```

Figure 4-3. Cont.

```
                         RECEIVE RSET COMMAND (RSET)                          PAGE    3

                    CROSS REFERENCE

LABEL      VALUE         REFERENCE

ADDBUF   B 0000           -21    50
EOM      P 0015            62   -64
FRMBIF   E 0003            17    60
GOFLG    E 0000            17    61
MEMORY   M 0000             0
NARG     ' 0000             0
NOTVAL   P 0027            67    70   -72
OPSTAT   E 0002            17    65    71
R        E 0001            17    64
RCV      E 0006            19    63    76
RDAFLG   E 0004            18    51    59
RDBUFF   E 0005            18    57
RSET     P 0000           -46
STACK    S 0000             0
```

Figure 4-3. Cont.

5.0 DELETE COMMAND OR RESPONSE I-FRAME

These two options limit the procedure to allow I frames to be commands only, or responses only, by deleting the I response and the I command respectively. This technique limits information frames to one direction for primary and secondary stations.

These options are treated together in this section, because the main effect of each is the same: one station loses the ability to transmit I-frames and the other loses the ability to receive them.

The SENDI process is used to transmit I-frames (Refer to Figure 5-1) both as commands and responses. The code for this process is presented in Figure 5-2. If I-frame transmission is deleted, the SENDI process is as shown in Figure 5-3. The 6800 code corresponding to the flow chart of Figure 5-3 is shown in Figure 5-4. The difference in number of instructions between these two routines is approximately 100 instructions. In addition to this reduction the CHICPNT routine can be deleted together with references to it in RR and RNR, removing an additional 60 instructions for a total of 160 instructions. Throughput can nearly be doubled if information transmission is limited to one direction based on the fact that the processor need manage half the number of buffers and pointers.

The flow chart for receiving I-frames is shown in Figure 5-5, and the corresponding code in Figure 5-6. If

I-frames are not to be received, this routine can be removed completely, saving approximately 75 instructions.

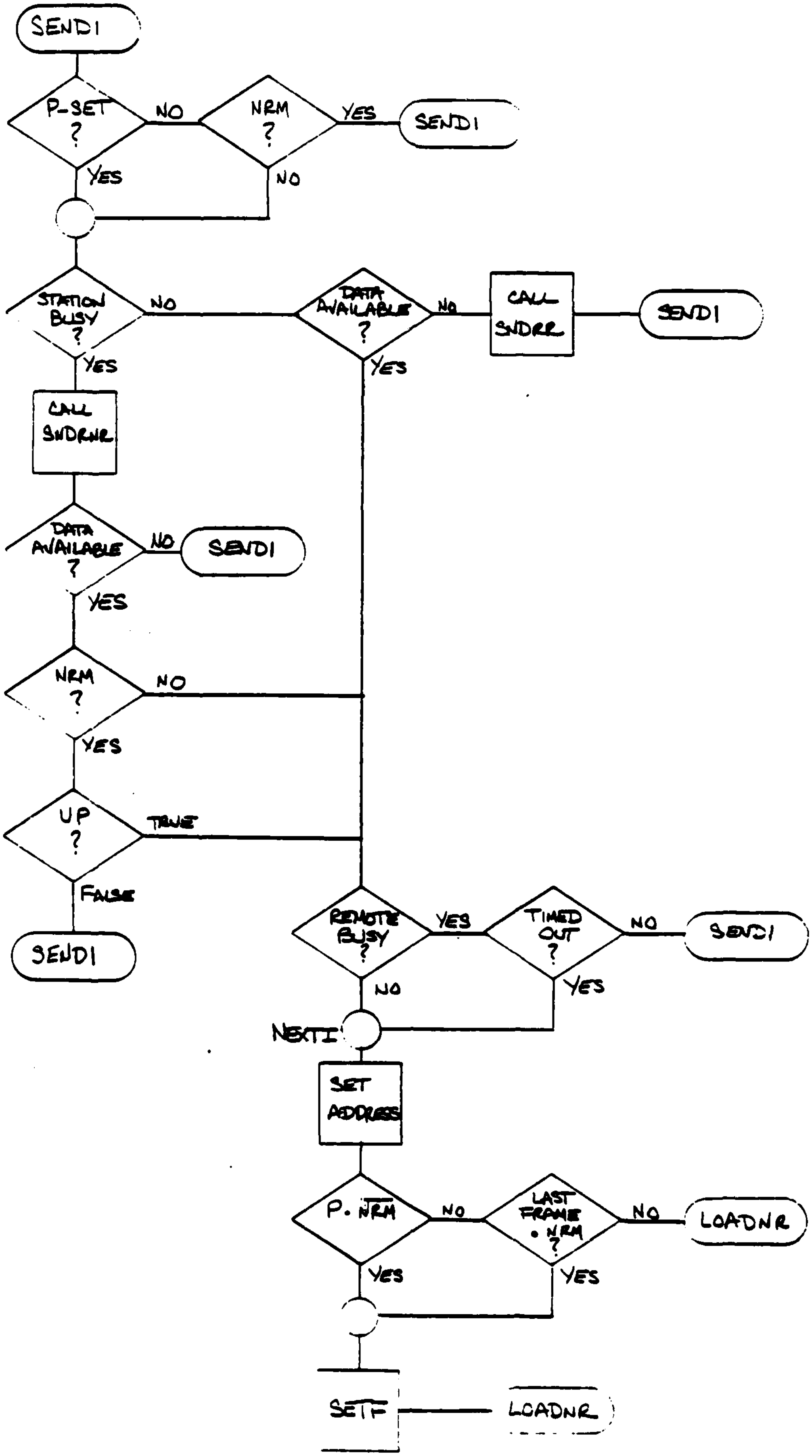


Figure 5-1. SENDI Process

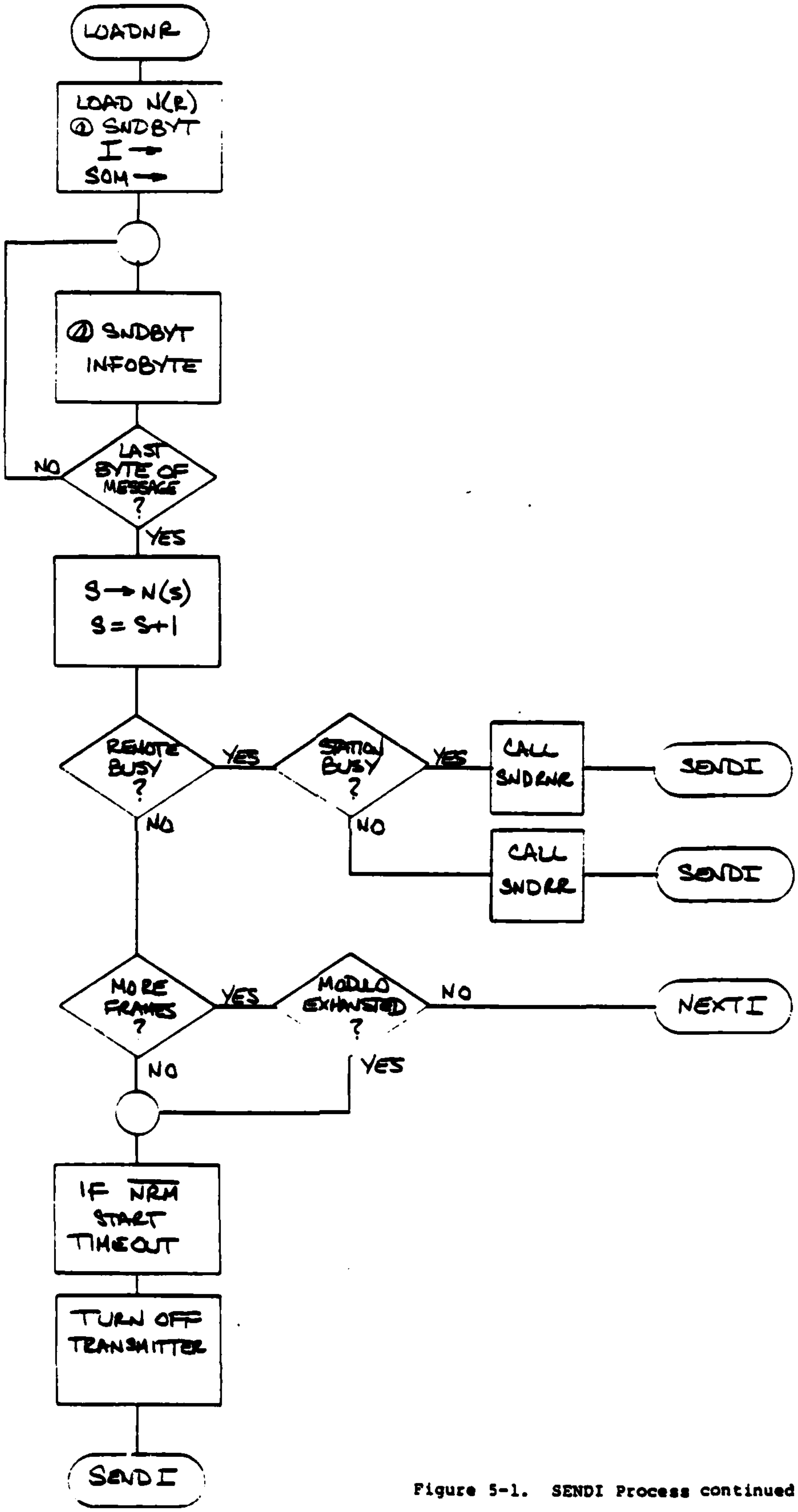


Figure 5-1. SENDI Process continued

```
ERR LINE  ADDR  B1 B2 B3 B4        SEND INFORMATION PROCESS (SENDI)                          PAGE    1

                                           TITLE   'SEND INFORMATION PROCESS (SENDI)'
                                           LIST    X
                                           NAME    SENDI
                                   *
                                   *** SENDI PROCESS
                                   *
                                   *   REFERENCES: PBIT,ITSMOD,STABUS,DAVAIL,REMBUS,LFRM,TSR
                                   *   MODIFIES:   CFIELD,FBIT,TCR,TDB,S,NS
                                   *
                                   *   CALLS:      SNDRR
                                   *               SNDRNR
                                   *   EXIT:       NONE
                                   *
                                   *   TRANSMITS I-FRAMES AND SUPERVISORY FRAMES AS REQUIRED.  IF DATA
                                   *   IS AVAILABLE FOR TRANSMISSION AND THE REMOTE STATION IS NOT
                                   *   BUSY, THE LOOP FOR TRANSMITTING I-FRAMES IS ENTERED AT NEXT I.
                                   *   THE COMPLETE I FRAME IS TRANSMITTED BYTE BY BYTE AND THE SEND
                                   *   VARIABLE (S) IS INCREMENTED.   IF THE REMOTE STATION IS FOUND
                                   *   TO BE BUSY, RR OR RNR IS TRANSMITTED AS APPROPRIATE.   IF MORE
                                   *   FRAMES ARE AVAILABLE FOR TRANSMISSION, THE LOOP IS REPEATED
                                   *   FROM NEXTI.   IF NOT, THE PROCESS IS REPEATED FROM SENDI.
                                   *

                                           XDEF    SENDI
                                           XREFB   CFIELD,PBIT,ITSMOD,STABUS,DAVAIL,REMBUS,LFRM
                                           XREFB   FBIT,TSR,TDB,TCR,S,NS
                                           XREF    SNDRR,SNDRNR

                                   *** SNDBYT MACRO DEFINITION
                                   *
                                   SNDBYT  MACRO   ONEBYT,TRFLG
                                           LOCAL   SETOK,SETOK1
                                           SEI                 PREVENT DATA CHANGE WHILE WRITING
                                   *
                                   *   TEST TRANSMITTER BUFFER EMPTY EVENT VARIABLE
                                   *
                                           LDAA    TSR
                                           ANDA    #$20        MASK OFF TBMT
                                           BNE     SETOK
                                   *        *
                                   *       SAVE PLACE AND JUMP TO WAIT PROCESS
                                   *        *
                                   SETOK   LDAA    ONEBYT
                                           STAA    TDB
                                           LDAA    #TRFLG
                                           BEQ     SETOK1
                                           STAA    TCR
                                   SETOK1  CLI
                                           ENDM
```

Figure 5-2. SENDI Process Code

```
ERR LINE  ADDR  B1 B2 B3 B4        SEND INFORMATION PROCESS (SENDI)                                  PAGE    2

     51                                     PSCT
     52  0000                       SENDI
     53                             *
     54                             *** BEGIN PROGRAM
     55                             *
     56  0000  7F 00 00       E             CLR     CFIELD
     57  0003  96 00          E             LDAA    PBIT       P-SET?
     58  0005  26 04                        BNE     CHKBUS
     59  0007  96 00          E             LDAA    ITSMOD     NRM?
     60  0009  27 F5                        BEQ     SENDI
     61  000B  96 00          E     CHKBUS  LDAA    STABUS     STATION BUSY?
     62  000D  27 11                        BEQ     DVAL       NO
     63  000F  BD 00 00       E             JSR     SNDRNR     YES
     64  0012  96 00          E             LDAA    DAVAIL     DATA AVAILABLE?
     65. 0014  27 EA                        BEQ     SENDI      NO
     66  0016  96 00          E             LDAA    ITSMOD     NRM?
     67  0018  26 E6                        BNE     SENDI      YES
     68  001A  96 00          E     CKRBUS  LDAA    REMBUS     REMOTE BUSY?
     69  001C  27 00                        BEQ     NEXTI      NO
     70                             *        *
     71                             *       IF TIMEOUT, NEXTI; IF NOT, SENDI
     72                             *        *
     73  001E  20 09                        BRA     NEXTI
     74  0020  96 00          E     DVAL    LDAA    DAVAIL     DATA AVAILABLE?
     75  0022  26 F6                        BNE     CKRBUS     YES
     76  0024  BD 00 00       E             JSR     SNDRR      NO
     77  0027  20 D7                        BRA     SENDI

     79                             *
     80                             *** SEND AN I-FRAME
     81                             *
     82  0029                       NEXTI
     83                             *        *
     84                             *       SET ADDRESS
     85                             *        *
     86  0029  96 00          E             LDAA    ITSMOD     NRM?
     87  002B  26 04                        BNE     CHKP       NO
     88  002D  96 00          E             LDAA    LFRM       YES; LAST FRAME?
     89  002F  27 0C                        BEQ     LOADNR     NO
     90  0031  96 00          E     CHKP    LDAA    PBIT       POLL BIT SET?
     91  0033  27 08                        BEQ     LOADNR     NO
     92  0035  97 00          E             STAA    FBIT       YES
     93  0037  96 00          E             LDAA    CFIELD
     94  0039  8A 10                        ORAA    #$10
     95  003B  97 00          E             STAA    CFIELD
     96  003D                       LOADNR
     97                             *        *
     98                             *       LOAD N(R), N(S) INTO CONTROL FIELD
     99                             *        *
    100  003D                               SNDBYT  CFIELD,#$80
    101  003D  0F                  +        SEI                PREVENT DATA CHANGE WHILE WRITING
    102                            +*
```

Figure 5-2. Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4       SEND INFORMATION PROCESS (SENDI)                           PAGE    3

     103                          **   TEST TRANSMITTER BUFFER EMPTY EVENT VARIABLE
     104                          **
     105  003E  96 00          E  *        LDAA    TSR
     106  0040  84 20             *        ANDA    #$20      MASK OFF TBMT
     107  0042  26 00             *        BNE     .00000
     108                          **        *
     109                          **       SAVE PLACE AND JUMP TO WAIT PROCESS
     110                          **        *
     111  0044  96 00          E  *.00000   LDAA    CFIELD
     112  0046  97 00          E  *        STAA    TDR
     113  0048  86 00             *        LDAA    #$00
     114  004A  27 02             *        BEQ     .00001
     115  004C  97 00          E  *        STAA    TCR
     116  004E  0E                *.00001  CLI
     117                           *        *
     118                           *        SEND INFO BYTES
     119                           *        *
     120  004F  96 00          E           LDAA    S
     121  0051  97 00          E           STAA    NS
     122  0053  4C                         INCA
     123  0054  97 00          E           STAA    S
     124  0056  96 00          E           LDAA    REMBUS    REMOTE BUSY?
     125  0058  27 0E                      BEQ     NORECK    NO
     126  005A  96 00          E           LDAA    STABUS    YES; STATION BUSY?
     127  005C  27 05                      BEQ     SNDR      NO
     128  005E  BD 00 00       E           JSR     SNDRNR    YES
     129  0061  20 9D                      BRA     SENDI
     130  0063  BD 00 00       E   SNDR    JSR     SNDRR
     131  0066  20 98                      BRA     SENDI
     132  0068                    NORECK
     133                           *        *
     134                           *        IF MORE FRAMES, NEXTI;
     135                           *        OTHERWISE TIMEOUT, TURN OFF TRANSMITTER AND SENDI
     136                           *        *
     137  0068                          END

 ASSEMBLER ERRORS =    0
```

Figure 5-2. Cont.

```
                    SEND INFORMATION PROCESS (SENDI)                    PAGE    4

                         CROSS REFERENCE

LABEL      VALUE         REFERENCE

CFIELD   E 0000              25    56    93    95   111
CHKBUS   P 0008              58   -61
CHKP     P 0031              87   -90
CKRBUS   P 001A             -68    75
DAVAIL   E 0004              25    64    74
DVAL     P 0020              62   -74
FBIT     E 0007              26    92
ITSMOD   E 0002              25    59    66    86
LFRN     E 0006              25    88
LOADMR   P 0030              89    91   -96
MEMORY   M 0000               0
MORECK   P 0060             125  -132
NARG       0000               0
NEXTI    P 0029              69    73   -82
NS       E 000C              26   121
PBIT     E 0001              25    57    90
REMBUS   E 0005              25    60   124
S        E 0000              26   120   123
SENDI    P 0000             -52    60    65    67    77   129   131
SNDR     P 0063             127  -130
SNDRMR   E 000E              27    63   120
SNDRR    E 000D              27    76   130
STABUS   E 0003              25    61   126
STACK    S 0000               0
TCR      E 000A              26   115
TDB      E 0009              26   112
TSR      E 000B              26   105
```

Figure 5-2. `Cont.

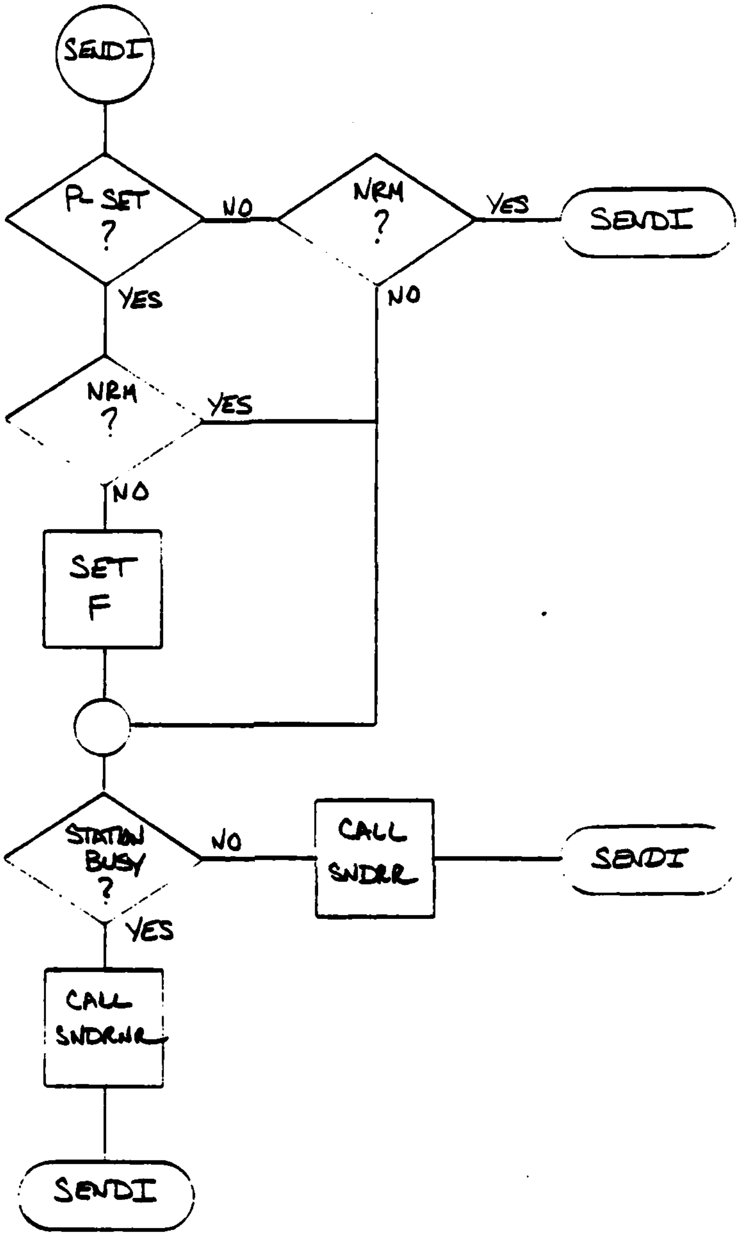


Figure 5-3. Delete I-Frame Transmission

```
ERR LINE  ADDR  B1 B2 B3 B4        DELETE RESPONSE I-FRAME TRANS (SENDI)                      PAGE    1

       1                                  TITLE   'DELETE RESPONSE I-FRAME TRANS (SENDI)'
       2                                  LIST    X
       3                                  NAME    SENDI
       4                           *
       5                           *** SENDI PROCESS
       6                           *
       7                           *   REFERENCES: PBIT,ITSNOD,STABUS
       8                           *   MODIFIES:   FBIT
       9                           *
      10                           *   CALLS:      SNDRR
      11                           *               SNDRNR
      12                           *   EXIT:       NONE
      13                           *
      14                           *   TRANSMITS SUPERVISORY FRAMES AS REQUIRED.
      15                           *

      17                                  XDEF    SENDI
      18                                  XREF    SNDRR,SNDRNR
      19                                  XREFB   PBIT,ITSNOD,FBIT,STABUS

      21                                  PSCT
      22  0000                     SENDI
      23                           *
      24                           *** BEGIN PROGRAM
      25                           *
      26  0000  96 00          E          LDAA    PBIT      P-SET?
      27  0002  27 08                     BEQ     CHKNRN    NO
      28  0004  D6 00          E          LDAB    ITSNOD    YES: NRN?
      29  0006  26 08                     BNE     CHKBUS    YES
      30  0008  97 00          E          STAA    FBIT      NO: SET FINAL BIT
      31  000A  20 04                     BRA     CHKBUS
      32  000C  D6 00          E   CHKNRN LDAB    ITSNOD    NRN?
      33  000E  26 F0                     BNE     SENDI     YES
      34  0010  96 00          E   CHKBUS LDAA    STABUS    STATION BUSY?
      35  0012  26 05                     BNE     SNDNR     YES
      36  0014  BD 00 00       E          JSR     SNDRR     NO
      37  0017  20 E7                     BRA     SENDI
      38  0019  BD 00 00       E   SNDNR  JSR     SNDRNR
      39  001C  20 E2                     BRA     SENDI
      40  001E                          END

 ASSEMBLER ERRORS =     0
```

Figure 5-4. Delete Response I Transmission Code

```
                              DELETE RESPONSE I-FRAME TRANS (SENDI)                PAGE    2

                         CROSS REFERENCE

LABEL       VALUE          REFERENCE

CHKBUS    P 0010             29      31     -34
CHKWRN    P 000C             27     -32
FBIT      E 0004             19      30
ITSMOD    E 0003             19      20      32
MEMORY    M 0000              0
MARG        0000              0
PBIT      E 0002             19      26
SENDI     P 0000            -22      33      37      39
SNDWR     P 0019             35     -38
SNDRWR    E 0001             18      38
SNDRR     E 0000             18      36
STABUS    E 0005             19      34
STACK     S 0000              0
```

Figure 5-4. Cont.

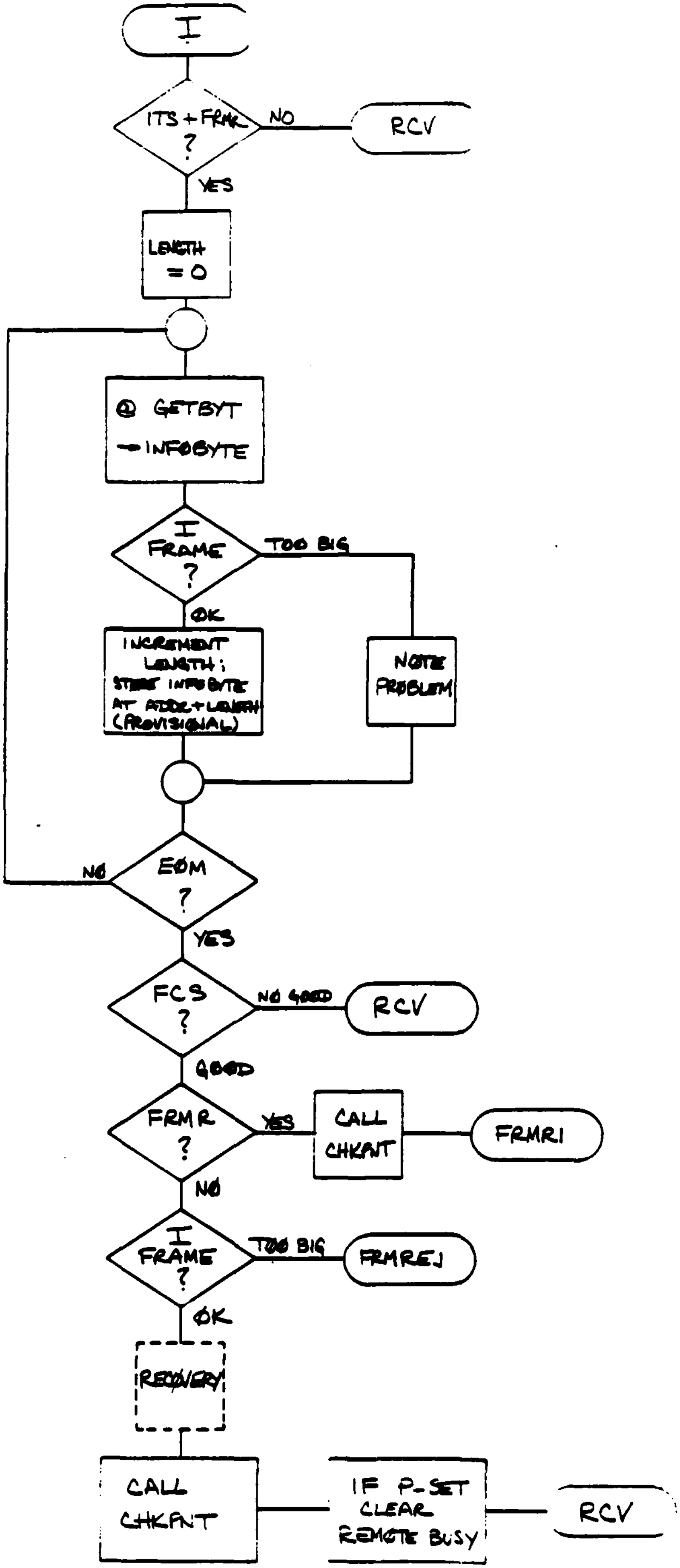


Figure 5-5.

Receive I Subroutine

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE INFORMATION FRAME (I)                          PAGE    1
                                           TITLE   'RECEIVE INFORMATION FRAME (I)'
                                           LIST    X
                                           NAME    I
                                   *
                                   *** RECEIVE I COMMAND-RESPONSE
                                   *
                                   *   REFERENCES:  GDFLG
                                   *                OPSTAT
                                   *                PBIT
                                   *                RDAFLG
                                   *                RDBUFF
                                   *                LENGNX
                                   *   MODIFIES:    FBIT
                                   *                INFBYT
                                   *                RENBUS
                                   *   CALLS:       RCV
                                   *                CHKPNT
                                   *                FRMREJ
                                   *                FRMR1
                                   *
                                   *   MAKES THE APPROPRIATE CHECKS ON THE I-FRAME CONTROL FIELD
                                   *   AND READS THE INFORMATION FIELD BYTE-BY-BYTE.

                                           XDEF    I
                                           XREF    RCV,CHKPNT,FRMREJ,FRMR1
                                           XREFB   OPSTAT,PBIT,FBIT,RDAFLG,RDBUFF,GDFLG
                                           XREFB   LENGNX,RENBUS
                                           BSCT
          0000                     LENGTH  RMB     1
          0001                     LENGFL  RMB     1
          0002                     INFBYT  RMB     1
                                   *
                                   *** GETBYT MACRO DEFINITION
                                   *
                                   GETBYT  MACRO   ONEBYT
                                           LOCAL   SETOK
                                           SEI             PREVENT DATA CHANGE WHILE READING
                                   *
                                   * TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
                                   *
                                           LDAA    RDAFLG
                                           BNE     SETOK
                                   *        *
                                   *   SAVE PLACE AND JUMP TO WAIT PROCESS
                                   *        *
                                   *
                                   SETOK   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
                                           STAA    ONEBYT
                                           CLR     RDAFLG
                                           CLI
                                           ENDM
```

Figure 5-6. Receive I Frame Code

```
ERR LINE  ADDR  B1 B2 B3 B4.        RECEIVE INFORMATION FRAME (I)                                PAGE    2

      53                                       PSCT
      54                              *
      55                              *** BEGIN PROGRAM
      56                              *
      57  0000                        I
      58  0000  96 00           E              LDAA    OPSTAT    CHECK OPERATIONAL STATE
      59  0002  2E 10                          BGT     CONT      ITS
      60  0004  8B 02                          ADDA    02
      61  0006  27 0C                          BEQ     CONT      FRMR
      62  0008  96 00           E              LDAA    PBIT      POLL BIT SET?
      63  000A  26 03                          BNE     PTWO      YES
      64  000C  7E 00 00        E              JMP     RCV
      65  000F  97 00           E     PTWO     STAA    FBIT
      66                              *         *
      67                              *        ACTIVATE TRANSMIT TASK TR-ON
      68                              *         *
      69  0011  7E 00 00        E              JMP     RCV
      70  0014  4F                    CONT     CLRA
      71  0015  97 00           B              STAA    LENGTH
      72  0017  97 01           B              STAA    LENGFL
      73                              *
      74                              *** RECEIVE MESSAGE LOOP
      75                              *
      76  0019                        LOOP     GETBYT  INFBYT
      77  0019  0F                   +         SEI               PREVENT DATA CHANGE WHILE READING
      78                             +*
      79                             +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
      80                             +*
      81  001A  96 00           E    +         LDAA    RDAFLG
      82  001C  26 00                +         BNE     .00000
      83                             +*         *
      84                             +*   SAVE PLACE AND JUMP TO WAIT PROCESS
      85                             +*         *
      86                             +*
      87  001E  96 00           E    +.00000   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
      88  0020  97 02           B    +         STAA    INFBYT
      89  0022  7F 00 00        E    +         CLR     RDAFLG
      90  0025  0E                   +         CLI
      91  0026  7D 00 00        E              TST     GOFLG
      92  0029  2C 03                          BGE     PTWO1     NORMAL OR EOM
      93  002B  7E 00 00        E              JMP     RCV       ERROR/EOM
      94  002E  2E 10                 PTWO1    BGT     CKFRMR    EOM/GOOD FRAME
      95  0030  D6 00           B              LDAB    LENGTH
      96  0032  D0 00           E              SUBB    LENGMX
      97  0034  2E 05                          BGT     TOOBIG
      98  0036  7C 00 00        B              INC     LENGTH
      99                              *         *
     100                              *        STORE INFBYT AT BASE ADDRESS+LENGTH
     101                              *         *
     102  0039  20 DE                          BRA     LOOP
     103  003B  7C 00 01        B     TOOBIG   INC     LENGFL
     104  003E  20 D9                          BRA     LOOP
     105  0040  96 00           E     CKFRMR   LDAA    OPSTAT
```

Figure 5-6. Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE INFORMATION FRAME (1)                              PAGE    3

     106  0042  8B 02                              ADDA     02
     107  0044  26 06                              BNE      CIFRM     NOT FRMR
     108  0046  BD 00 00       E                   JSR      CHKPNT
     109  0049  7E 00 00       E                   JMP      FRMR1
     110  004C  7D 00 01       B    CIFRM          TST      LENGFL    I-FRAME TOO BIG?
     111  004F  27 03                              BEQ      PERREC    NO
     112  0051  7E 00 00       E                   JMP      FRMREJ    YES
     113  0054                      PERREC
     114                            *               *
     115                            *              PERFORM RECOVERY
     116                            *               *
     117  0054  BD 00 00       E                   JSR      CHKPNT
     118  0057  7D 00 00       E                   TST      PBIT
     119  005A  26 03                              BNE      CLRBUS
     120  005C  7E 00 00       E                   JMP      RCV
     121  005F  7F 00 00       E    CLRBUS         CLR      RENBUS
     122  0062  7E 00 00       E                   JMP      RCV
     123  0065                                 END

  ASSEMBLER ERRORS =    0
```

Figure 5-6. Cont.

```
                              RECEIVE INFORMATION FRAME (I)                                     PAGE    4

                           CROSS REFERENCE

LABEL      VALUE          REFERENCE

CHKPNT   E 0001             25   100   117
CIFRN    P 004C            107  -110
CKFRMR   P 0040             94  -105
CLRBUS   P 005F            119  -121
CONT     P 0014             59    61   -70
FBIT     E 0006             26    65
FRMRI    E 0003             25   109
FRMREJ   E 0002             25   112
GBFLG    E 0009             26    91
I        P 0000            -57
INFBYT   B 0002            -31    80
LENGFL   B 0001            -30    72   103   110
LENGMX   E 000A             27    96
LENGTH   B 0000            -29    71    95    98
LOOP     P 0019            -76   102   104
MEMORY   M 0000              0
NARG       0000              0
OPSTAT   E 0004             26    58   105
PBIT     E 0005             26    62   110
PERREC   P 0054            111  -113
PTUG     P 000F             63   -65
PTUG1    P 002E             92   -94
RCV      E 0000             25    64    69    93   120   122
RDAFLG   E 0007             26    81    89
RDBUFF   E 0000             26    87
REMBUS   E 000B             27   121
STACK    S 0000              0
TOOBIG   P 003B             97  -103
```

Figure 5-6. Cont.

6.0 UNNUMBERED POLLING FUNCTION

Option 6 provides for the ability to perform unnumbered group polling as well as unnumbered individual polling by the addition of the UP command. The UP command is used to solicit a response frame from a single station, or from a group of stations, by establishing a logical operational condition that exists at each addressed station for one respond opportunity.

The flow chart for the reception of the unnumbered polling command and the corresponding 6800 assembly language code are presented in Figures 6-1 and 6-2 respectively. Approximately 20 instructions are required for this routine. The receive UP function also requires some minor modification to the SENDI process and the processes for sending Receive Ready/Receive Not Ready. These modifications are shown in the flow charts of Figures 6-3 and 6-4. Effects on throughput are difficult ot judge at this level of implementation.

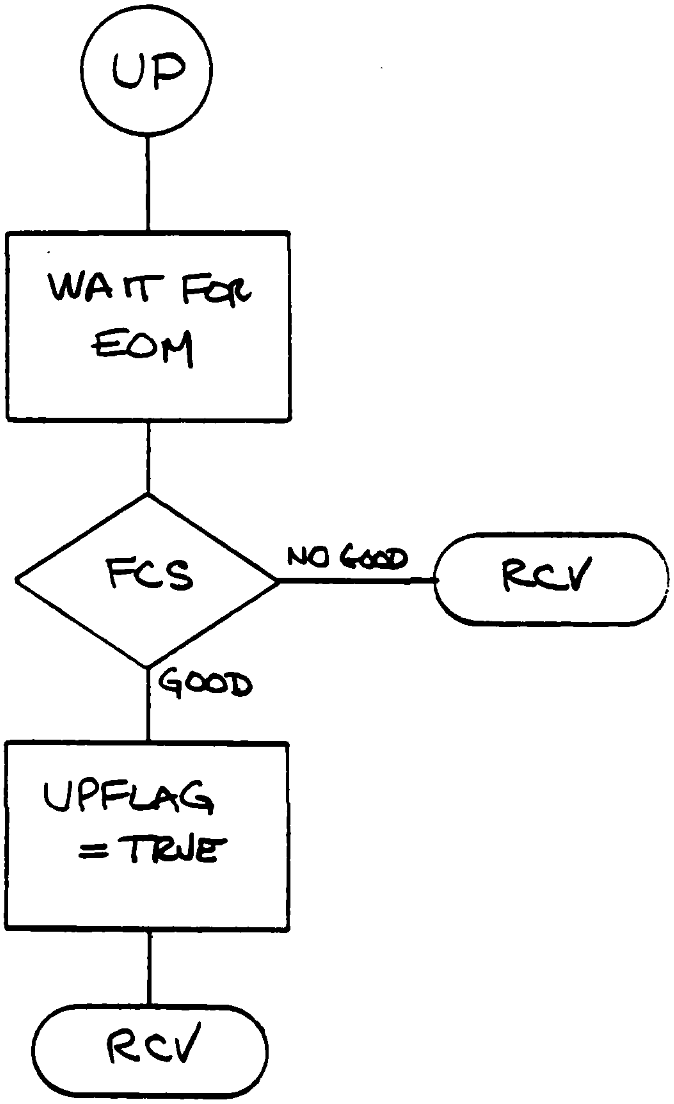


Figure 6-1. Receive UP Command

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE UNNUMBERED POLLING COMMAND (UP)                    PAGE    1
       1                                   TITLE   'RECEIVE UNNUMBERED POLLING COMMAND (UP)'
       2                                   LIST    X
       3                                   NAME    UP
       4                            *
       5                            *** RECEIVE UP COMMAND
       6                            *
       7                            *   REFERENCES:  GDFLG
       8                            *                RDAFLG
       9                            *                RDBUFF
      10                            *   MODIFIES:    UPFLAG
      11                            *
      12                            *   THIS ROUTINE SETS THE UPFLAG, INDICATING THAT A UP COMMAND
      13                            *   HAS BEEN RECEIVED.

      15                                   XDEF     UP
      16                                   XREF    RCV
      17                                   XREFB   RDAFLG,RDBUFF,GDFLG,UPFLAG
      18                                   DSCT
      19  0000                      ADDBUF  RMB     1
      20                            *
      21                            *** GETBYT MACRO DEFINITION
      22                            *
      23                            GETBYT  MACRO   ONEBYT
      24                                    LOCAL   SETOK
      25                                    SEI                   PREVENT DATA CHANGE WHILE READING
      26                            *
      27                            * TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
      28                            *
      29                                    LDAA    RDAFLG
      30                                    BNE     SETOK
      31                            *       *
      32                            *   SAVE PLACE AND JUMP TO WAIT PROCESS
      33                            *       *
      34                            *
      35                            SETOK   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
      36                                    STAA    ONEBYT
      37                                    CLR     RDAFLG
      38                                    CLI
      39                                    ENDM
```

Figure 6-2. Receive UP Command Code

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE UNNUMBERED POLLING COMMAND (UP)                   PAGE    2

     41                                      PSCT
     42                            *
     43                            *    WAIT FOR EOM
     44                            *
     45  0000                      UP        GETBYT  ADDBUF
     46  0000  0F                  +         SEI                 PREVENT DATA CHANGE WHILE READING
     47                            +*
     48                            +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
     49                            +*
     50  0001  96 00            E  +         LDAA    RDAFLG
     51  0003  26 00               +         BNE     .00000
     52                            +*         *
     53                            +*    SAVE PLACE AND JUMP TO WAIT PROCESS
     54                            +*         *
     55                            +*
     56  0005  96 00            E  +.00000    LDAA    RDBUFF     STORE RDBUFF IN MACRO ARG
     57  0007  97 00            E  +         STAA    ADDBUF
     58  0009  7F 00 00         E  +         CLR     RDAFLG
     59  000C  0E                  +         CLI
     60  000D  7D 00 00         E            TST     GBFLG
     61  0010  2E 03                         BGT     EOM       HAVE EOM
     62  0012  7E 00 00         E            JMP     RCV       ERROR
     63  0015  7F 00 00         E  EOM       CLR     UPFLAG
     64  0018  7C 00 00         E            INC     UPFLAG
     65  001B  7E 00 00         E            JMP     RCV
     66  001E                            END

  ASSEMBLER ERRORS =    0
```

Figure 6-2 Cont.

```
                                RECEIVE UNNUMBERED POLLING COMMAND (UP)                PAGE    3

                          CROSS REFERENCE

LABEL      VALUE         REFERENCE

ADDBUF   B 0000            -19    57
EOM      P 0015             61   -63
GOFLG    E 0003             17    60
MEMORY   M 0000              0
MARG       0000              0
RCV      E 0000             16    62    65
RDAFLG   E 0001             17    50    58
RDBUFF   E 0002             17    56
STACK    S 0000              0
UP       P 0000            -45
UPFLAG   E 0004             17    63    64
```

Figure 6-2. Cont.

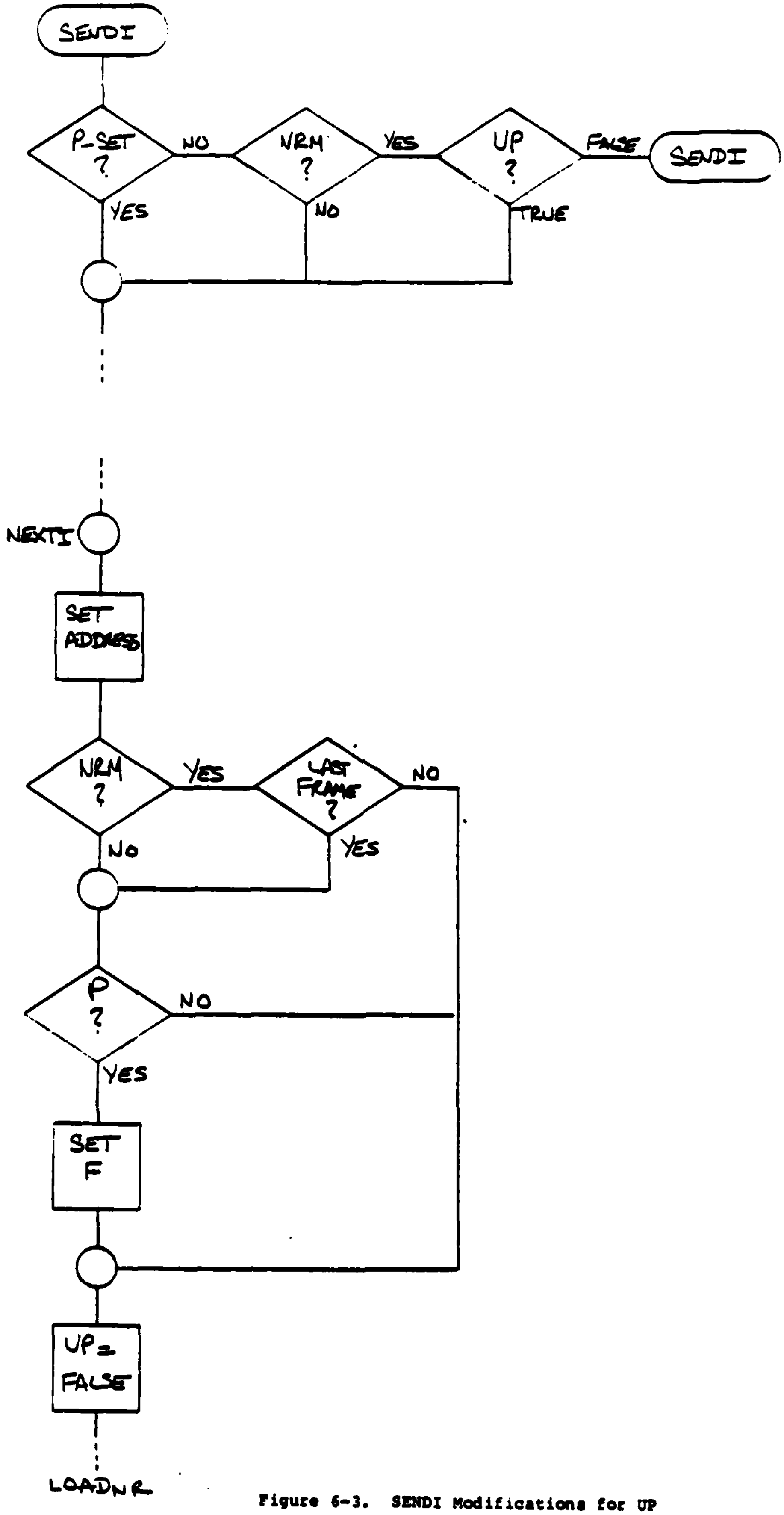


Figure 6-3. SENDI Modifications for UP

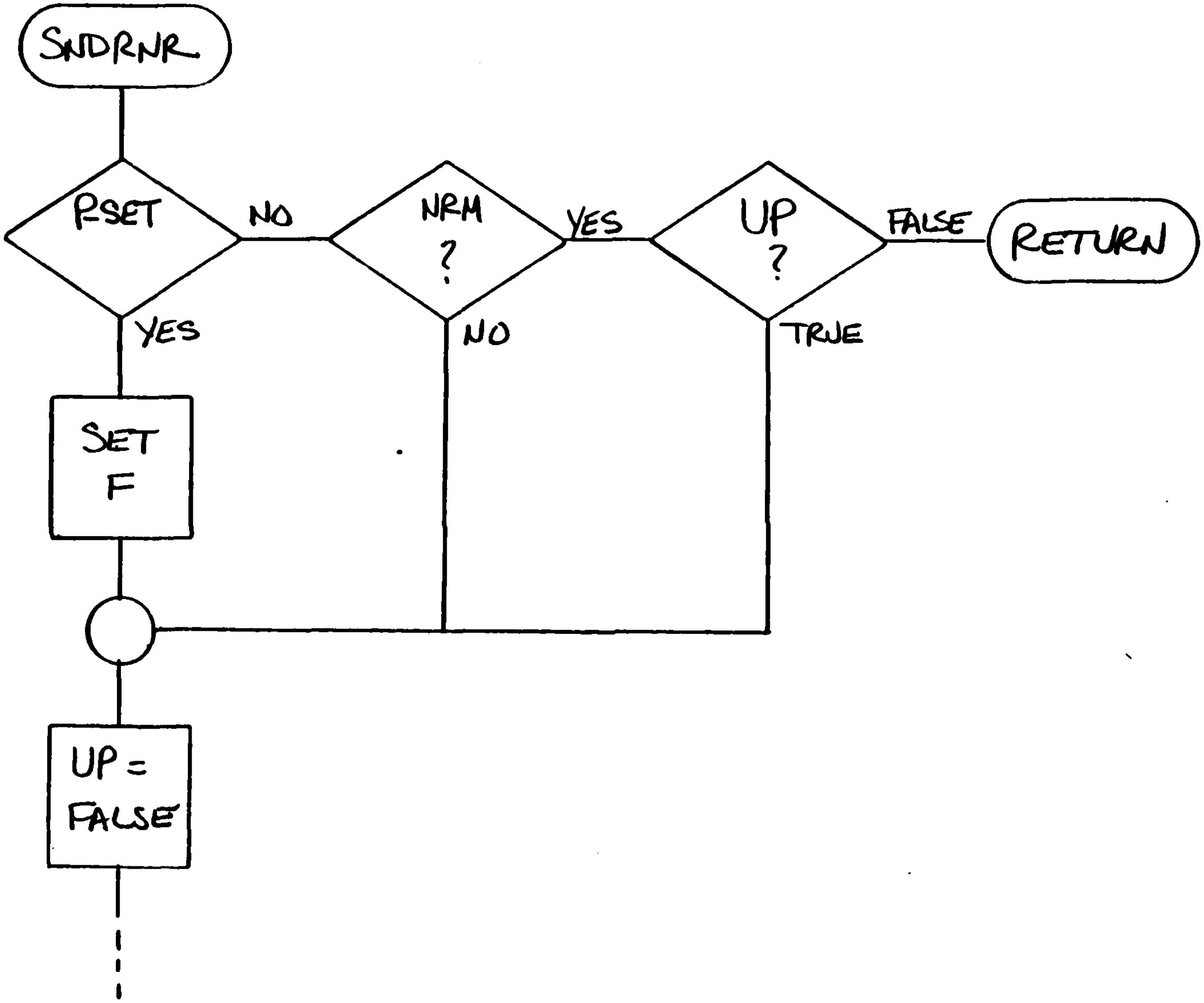


Figure 6-4. SNDRNR Modifications for UP

7.0 INITIALIZATION FUNCTION

This option provides the ability to initialize remote stations and the ability to request initialization. The SIM command and the RIM response are added. The SIM command is used to request a remote station to initiate a station-specified procedure to initialize its link-level control function. The RIM response is used by the Secondary/ combined station to request the SIM command.

The flow chart for the reception of the SIM command and the corresponding 6800 code are given in Figures 7-1 and 7-2. The flow chart for the transmission of the RIM response and corresponding code are given in Figures 7-3 and 7-4. Some modification to the module that determines the operational state is required to accommodate the initialization state and the RIM condition. This module is used at the beginning of the received command handlers for example, it appears in the received I-frame. The modifications to this routine are shown in Figure 7-5.

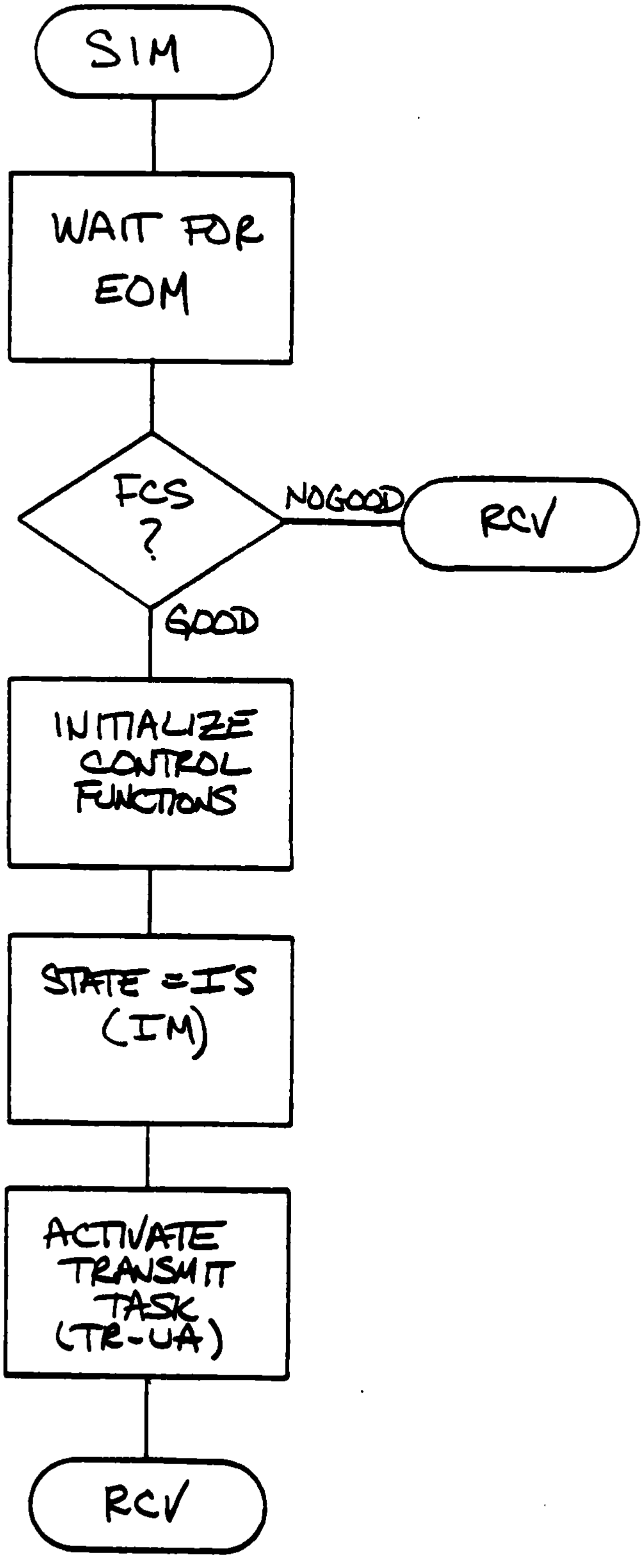


Figure 7-1. Receive SIM Command

```
ERR LINE  ADDR  B1 B2 B3 B4       RECEIVE SET INITIALIZATION MODE COMMAND (SIN)             PAGE     1

       1                                 TITLE   'RECEIVE SET INITIALIZATION MODE COMMAND (SIN)'
       2                                 LIST    X
       3                                 NAME    SIN
       4                          *
       5                          *** RECEIVE SIN COMMAND
       6                          *
       7                          *   REFERENCES:  GOFLG
       8                          *                RDAFLG
       9                          *                RDBUFF
      10                          *   MODIFIES:    OPSTAT
      11                          *
      12                          *   THIS ROUTINE INITIALIZES CONTROL FUNCTIONS, SETS OPERATIONAL
      13                          *   STATE TO IS, AND SENDS WA.

      15                                 XDEF     SIN
      16                                 XREF    RCV
      17                                 XREFB   RDAFLG,RDBUFF,GOFLG,OPSTAT
      18                                 DSCT
      19  0000                    ADDBUF  RMB     1
      20                          *
      21                          *** GETBYT MACRO DEFINITION
      22                          *
      23                          GETBYT  MACRO   ONEBYT
      24                                  LOCAL   SETOK
      25                                  SEI                   PREVENT DATA CHANGE WHILE READING
      26                          *
      27                          * TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
      28                          *
      29                                  LDAA    RDAFLG
      30                                  BNE     SETOK
      31                          *        *
      32                          *   SAVE PLACE AND JUMP TO WAIT PROCESS
      33                          *        *
      34                          *
      35                          SETOK   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
      36                                  STAA    ONEBYT
      37                                  CLR     RDAFLG
      38                                  CLI
      39                                  ENDM
```

Figure 7-2. Receive SIM Command Code

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE SET INITIALIZATION MODE COMMAND (SIN)              PAGE    2

     41                                        PSCT
     42                              *
     43                              *   WAIT FOR EON
     44                              *
     45  0000                        SIN       GETBYT  ADDBUF
     46  0000  0F                   +          SEI                 PREVENT DATA CHANGE WHILE READING
     47                             +*
     48                             +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
     49                             +*
     50  0001  96 00            E   +          LDAA    RDAFLG
     51  0003  26 00                +          BNE     .00000
     52                             +*          *
     53                             +*   SAVE PLACE AND JUMP TO WAIT PROCESS
     54                             +*          *
     55                             +*
     56  0005  96 00            E   +.00000   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
     57  0007  97 00            D   +          STAA    ADDBUF
     58  0009  7F 00 00         E   +          CLR     RDAFLG
     59  000C  0E                   +          CLI
     60  000D  7D 00 00         E              TST     GOFLG
     61  0010  2E 03                           BGT     EON      HAVE EON
     62  0012  7E 00 00         E              JMP     RCV      ERROR
     63  0015                        EON
     64                              *          *
     65                              *          INITIALIZE CONTROL FUNCTIONS
     66                              *          *
     67  0015  86 FF                           LDAA    #-1
     68  0017  97 00            E              STAA    OPSTAT
     69                              *          *
     70                              *          ACTIVATE TRANSMIT TASK (TR-WA)
     71                              *          *
     72  0019  7E 00 00         E              JMP     RCV
     73  001C                             END

  ASSEMBLER ERRORS =    0
```

Figure 7-2 Cont.

```
                              RECEIVE SET INITIALIZATION MODE COMMAND (SIN)             PAGE    3

                         CROSS REFERENCE

LABEL     VALUE        REFERENCE

ADDBUF  B 0000           -19    57
EON     P 0015            61   -63
EOFLG   E 0003            17    60
MEMORY  N 0000             0
NARG      0000             0
OPSTAT  E 0064            17    60
RCV     E 0000            16    62    72
RDAFLG  E 0001            17    50    58
RDBUFF  E 0002            17    56
SIN     P 0000           -45
STACK   S 0000             0
```

Figure 7-2 Cont.

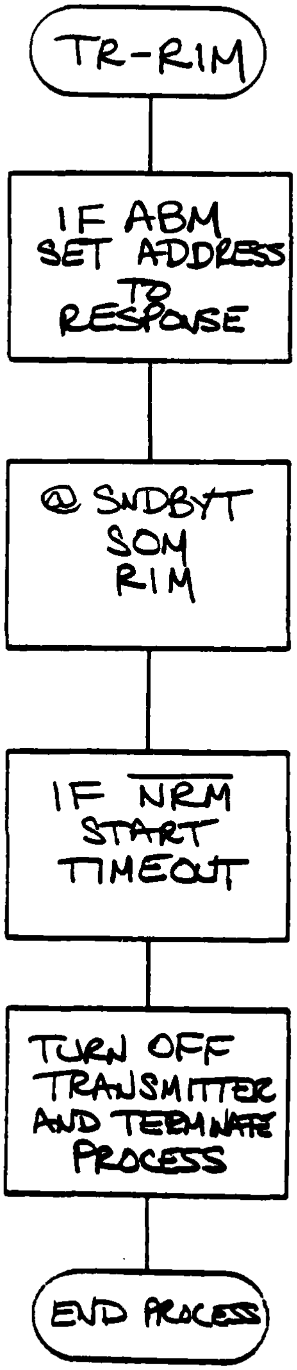

TR-RIM
IF ABM SET ADDRESS TO RESPONSE
@ SNDBYT SOM RIM
IF $\overline{\text{NRM}}$ START TIMEOUT
TURN OFF TRANSMITTER AND TERMINATE PROCESS
END PROCESS

Figure 7-3. Transmit RIM Response

```
ERR LINE  ADDR  B1 B2 B3 B4        TRANSMIT REQUEST INITIALIZATION MODE RESPONSE (RIN         PAGE    1

       1                                  TITLE   'TRANSMIT REQUEST INITIALIZATION MODE RESPONSE (RIN)'
       2                                  LIST    X
       3                                  NAME    RIN
       4                           *
       5                           *** TRANSMIT RIN RESPONSE
       6                           *
       7                           *   REFERENCES: TSR
       8                           *   MODIFIES:   CFIELD
       9                           *               TDB
      10                           *               TCR
      11                           *

      13                                  XDEF    RIN
      14                                  XREFB   TSR,CFIELD,TDB,TCR

      16                           *** SNDBYT MACRO DEFINITION
      17                           *
      18                           SNDBYT  MACRO   ONEBYT,TRFLG
      19                                   LOCAL   SETOK,SETOK1
      20                                   SEI              PREVENT DATA CHANGE WHILE WRITING
      21                           *
      22                           *   TEST TRANSMITTER BUFFER EMPTY EVENT VARIABLE
      23                           *
      24                                   LDAA    TSR
      25                                   ANDA    #$20       MASK OFF TDRE
      26                                   BNE     SETOK
      27                           *        *
      28                           *       SAVE PLACE AND JUMP TO WAIT PROCESS
      29                           *        *
      30                           SETOK   LDAA    #ONEBYT
      31                                   STAA    TDB
      32                                   LDAA    #TRFLG
      33                                   BEQ     SETOK1
      34                                   STAA    TCR
      35                           SETOK1  CLI
      36                                   ENDM
```

Figure 7-4. Transmit RIM Command Code

```
ERR LINE  ADDR  B1 B2 B3 B4        TRANSMIT REQUEST INITIALIZATION MODE RESPONSE (RIN        PAGE    2

     38                                      PSCT
     39  0000                          RIN
     40                                *
     41                                *** BEGIN PROGRAM
     42                                *
     43                                *         *
     44                                *         IF ABN, SET ADDRESS TO RESPONSE
     45                                *         *
     46  0000  96 00          E                  LDAA    CFIELD
     47  0002  84 10                             ANDA    #$10
     48  0004  8A 07                             ORAA    #$07
     49  0006  97 00          E                  STAA    CFIELD
     50  0008                                    SNDBYT  CFIELD,#$0
     51  0008  0F                       *        SEI                 PREVENT DATA CHANGE WHILE WRITING
     52                                **
     53                                **   TEST TRANSMITTER BUFFER EMPTY EVENT VARIABLE
     54                                **
     55  0009  96 00          E         *        LDAA    TSR
     56  000B  84 20                    *        ANDA    #$20       MASK OFF TBMT
     57  000D  26 00                    *        BNE     .00000
     58                                **         *
     59                                **        SAVE PLACE AND JUMP TO WAIT PROCESS
     60                                **         *
     61  000F  96 00          E         *.00000   LDAA    CFIELD
     62  0011  97 00          E         *        STAA    TDR
     63  0013  86 00                    *        LDAA    #$00
     64  0015  27 02                    *        BEQ     .00001
     65  0017  97 00          E         *        STAA    TCR
     66  0019  0E                       *.00001  CLI
     67                                 *         *
     68                                 *        IF NOT NRM, START TIMEOUT
     69                                 *        TURN OFF TRANSMITTER & TERMINATE PROCESS
     70                                 *         *
     71  001A                                 END

  ASSEMBLER ERRORS =    0
```

Figure 7-4 Cont.

```
                              TRANSMIT REQUEST INITIALIZATION MODE RESPONSE (RIN          PAGE    3

                         CROSS REFERENCE

LABEL      VALUE          REFERENCE

CFIELD   E 0001              14     46     49     61
MEMORY   M 0000               0
MARG       0000               0
RIN      P 0000             -39
STACK    S 0000               0
TCR      E 0003              14     65
TDB      E 0002              14     62
TSR      E 0000              14     55
```

Figure 7-4 Cont.

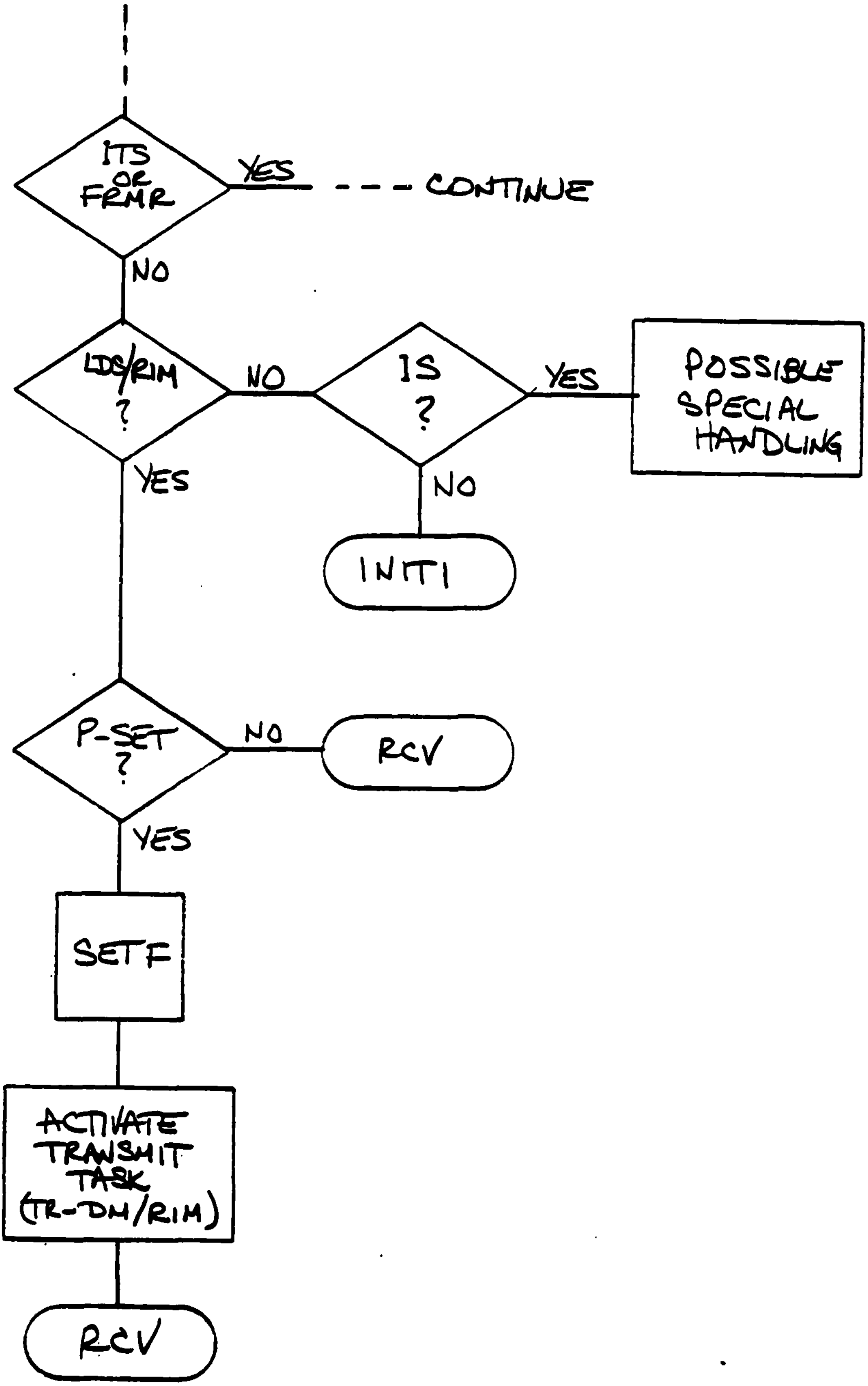


Figure 7-5. Modification for SIM/RIM

8.0 UNNUMBERED INFORMATION FUNCTION

This option provides the ability to exchange information fields without impacting the send and receive variables, and provides for the addition of the UI command and the UI response. Since the frame is not sequence number verified, the frame may be lost or duplicated if a link exception condition occurs.

The UI function is very similar to the transmit I function, assuming that a message is a number of bytes. The UI function requires no error recovery based on send and receive variables nor buffering and pointers for multiple frames. A flow chart for receiving UI and the corresponding 6800 code are given in Figures 8-1 and 8-2. Comparing Figure 8-2 with Figure 5-6 reveals the difference in code required to send a UI frame instead of an I frame. Of course, a UI frame may be sent in addition to an I frame.

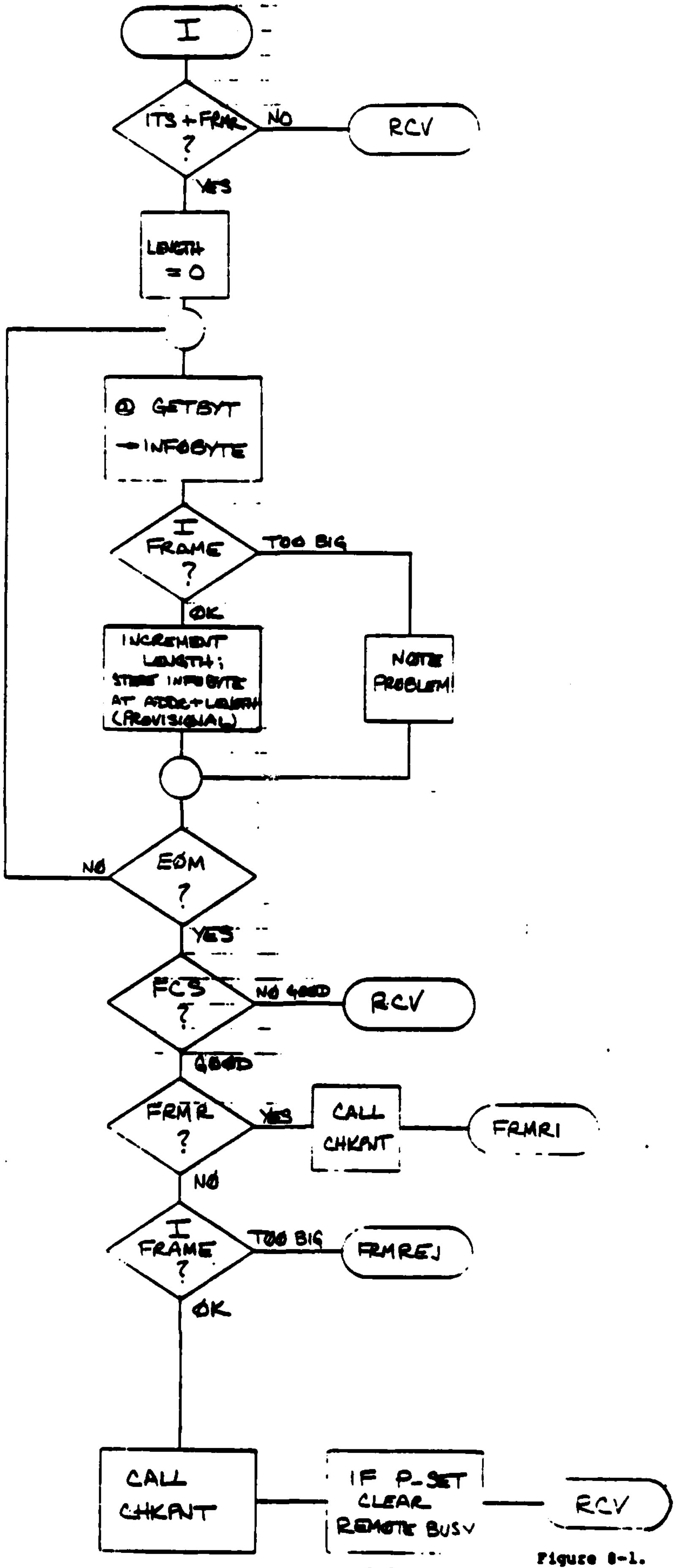


Figure 8-1. Receive UI Subroutine

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE UNNUMBERED INFORMATION FRAME (UI)                  PAGE    1

      1                                    TITLE   'RECEIVE UNNUMBERED INFORMATION FRAME (UI)'
      2                                    LIST    X
      3                                    NAME    UI
      4                                *
      5                                *** RECEIVE UI COMMAND-RESPONSE
      6                                *
      7                                *   REFERENCES:  GBFLG
      8                                *                OPSTAT
      9                                *                PBIT
     10                                *                RDAFLG
     11                                *                RDBUFF
     12                                *                LENGMX
     13                                *   MODIFIES:    FBIT
     14                                *                INFBYT
     15                                *                REMBUS
     16                                *   CALLS:       RCV
     17                                *                CHKPNT
     18                                *                FRMREJ
     19                                *                FRMRI
     20                                *
     21                                *   READS THE INFORMATION FIELD BYTE-BY-BYTE.

     23                                    XDEF     UI
     24                                    XREF    RCV,CHKPNT,FRMREJ,FRMRI
     25                                    XREFB   OPSTAT,PBIT,FBIT,RDAFLG,RDBUFF,GBFLG
     26                                    XREFB   LENGMX,REMBUS
     27                                    BSCT
     28  0000                          LENGTH  RMB     1
     29  0001                          LENGFL  RMB     1
     30  0002                          INFBYT  RMB     1
     31                                *
     32                                *** GETBYT MACRO DEFINITION
     33                                *
     34                                GETBYT  MACRO   ONEBYT
     35                                        LOCAL   SETOK
     36                                        SEI             PREVENT DATA CHANGE WHILE READING
     37                                *
     38                                * TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
     39                                *
     40                                        LDAA    RDAFLG
     41                                        BNE     SETOK
     42                                *         *
     43                                *   SAVE PLACE AND JUMP TO WAIT PROCESS
     44                                *         *
     45                                *
     46                                SETOK   LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
     47                                        STAA    ONEBYT
     48                                        CLR     RDAFLG
     49                                        CLI
     50                                        ENDM
```

Figure 8-2 Receive UI Frame Code

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE UNNUMBERED INFORMATION FRAME (UI)                 PAGE    2

     52                                     PSCT
     53                            *
     54                            *** BEGIN PROGRAM
     55                            *
     56   0000                     UI
     57   0000  96 00        E              LDAA    OPSTAT    CHECK OPERATIONAL STATE
     58   0002  2E 10                       BGT     CONT      ITS
     59   0004  8B 02                       ADDA    02
     60   0006  27 0C                       BEQ     CONT      FRMR
     61   0008  96 00        E              LDAA    PBIT      POLL BIT SET?
     62   000A  26 03                       BNE     PTW0      YES
     63   000C  7E 00 00     E              JMP     RCV
     64   000F  97 00        E     PTW0     STAA    FBIT
     65                            *         *
     66                            *        ACTIVATE TRANSMIT TASK TR-ON
     67                            *         *
     68   0011  7E 00 00     E              JMP     RCV
     69   0014  4F                 CONT     CLRA
     70   0015  97 00        B              STAA    LENGTH
     71   0017  97 01        B              STAA    LENGFL
     72                            *
     73                            *** RECEIVE MESSAGE LOOP
     74                            *
     75   0019                     LOOP     GETBYT  INFBYT
     76   0019  0F                +         SEI                PREVENT DATA CHANGE WHILE READING
     77                           +*
     78                           +* TEST RECEIVE DATA AVAILABLE EVENT VARIABLE
     79                           +*
     80   001A  96 00        E    +         LDAA    RDAFLG
     81   001C  26 00             +         BNE     .00000
     82                           +*         *
     83                           +*   SAVE PLACE AND JUMP TO WAIT PROCESS
     84                           +*         *
     85                           +*
     86   001E  96 00        E    +.00000    LDAA    RDBUFF    STORE RDBUFF IN MACRO ARG
     87   0020  97 02        B    +         STAA    INFBYT
     88   0022  7F 00 00     E    +         CLR     RDAFLG
     89   0025  0E                +         CLI
     90   0026  7D 00 00     E              TST     GOFLG
     91   0029  2C 03                       BGE     PTW01     NORMAL OR EOM
     92   002B  7E 00 00     E              JMP     RCV       ERROR/EOM
     93   002E  2E 10              PTW01    BGT     CKFRMR    EOM/GOOD FRAME
     94   0030  D6 00        B              LDAB    LENGTH
     95   0032  D0 00        E              SUBB    LENGMX
     96   0034  2E 05                       BGT     TOOBIG
     97   0036  7C 00 00     B              INC     LENGTH
     98                            *         *
     99                            *        STORE INFBYT AT BASE ADDRESS+LENGTH
    100                            *         *
    101   0039  20 DE                       BRA     LOOP
    102   003B  7C 00 01     B     TOOBIG   INC     LENGFL
    103   003E  20 D9                       BRA     LOOP
    104   0040  96 00        E     CKFRMR   LDAA    OPSTAT
```

Figure B-2 Cont.

```
ERR LINE  ADDR  B1 B2 B3 B4        RECEIVE UNNUMBERED INFORMATION FRAME (UI)                PAGE   3

     105  0042  8B 02                            ADDA    02
     106  0044  26 06                            BNE     CIFRM     NOT FRMR
     107  0046  BD 00 00      E                  JSR     CHKPNT
     108  0049  7E 00 00      E                  JMP     FRMR1
     109  004C  7D 00 01      D     CIFRM        TST     LENGFL    I-FRAME TOO BIG?
     110  004F  27 03                            BEQ     PERREC    NO
     111  0051  7E 00 00      E                  JMP     FRMREJ    YES
     112  0054                      PERREC
     113  0054  BD 00 00      E                  JSR     CHKPNT
     114  0057  7D 00 00      E                  TST     PBIT
     115  005A  26 03                            BNE     CLRBUS
     116  005C  7E 00 00      E                  JMP     RCV
     117  005F  7F 00 00      E     CLRBUS       CLR     REMBUS
     118  0062  7E 00 00      E                  JMP     RCV
     119  0065                             END

  ASSEMBLER ERRORS =    0
```

Figure B-2 Cont.

```
                                CROSS REFERENCE

LABEL      VALUE            REFERENCE

CHKPNT   E 0001               24   107   113
CIFRM    P 004C              106  -109
CKFRMR   P 0040               93  -104
CLRBUS   P 005F              115  -117
CONT     P 0014               58    60   -69
FBIT     E 0006               25    64
FRMR1    E 0003               24   100
FRMREJ   E 0002               24   111
GBFLG    E 0009               25    90
INFBYT   B 0002              -30    87
LENGFL   B 0001              -29    71   102   109
LENGMX   E 000A               26    95
LENGTH   B 0000              -28    78    94    97
LOOP     P 0019              -75   101   103
MEMORY   M 0000                0
MARG       0000                0
OPSTAT   E 0004               25    57   104
PBIT     E 0005               25    61   114
PERREC   P 0054              110  -112
PTUB     P 000F               62   -64
PTUB1    P 002E               91   -93
RCV      E 0000               24    63    68    92   116   118
RDAFLG   E 0007               25    88    88
RDBUFF   E 0008               25    86
REMBUS   E 0008               26   117
STACK    S 0000                0
TOOBIG   P 0030               96  -102
UI       P 0000              -56
```

Figure B-2 Cont.

9.0 REFERENCES

1. "Use of a Microprocessor to Implement an ADCCP Protocol with Reject and Selective Reject" (Federal Standard 1003) "Delta Information Systems, Inc. August 1981.

2. "Use of a Microprocessor To Implement an ADCCP Protocal (Federal Standard 1003)" Delta Information Systems, Inc. July 1980.

3. Malcon Easton, "Batch Throughput Efficiency of ADCCP/ HDLC/SDLC Selective Reject Protocols" Data Communications pp 187-195, February 1980.